UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

ANELISE LEMKE KOLOGESKI

Combinação de Estratégias para Tolerar Falhas em Interconexões e Aumentar o Rendimento na Produção de Redes Intra-Chip

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Microeletrônica

Prof. Drª. Fernanda Gusmão de Lima Kastensmidt
Orientadora

Porto Alegre, Dezembro de 2011.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Kologeski, Anelise Lemke

Combinação de Estratégias para Tolerar Falhas em Interconexões e Aumentar o Rendimento na Produção de Redes Intra-Chip / Anelise Lemke Kologeski – Porto Alegre: Programa de Pós-Graduação em Microeletrônica, Dezembro de 2011.

107 p.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica. Porto Alegre, BR – RS, 2011. Orientadora: Fernanda Gusmão de Lima Kastensmidt.

1.Redes Intra-Chip. 2.Tolerância a Falhas 3.Microeletrônica. I. Kastensmidt, Fernanda Gusmão de Lima II. Combinação de Estratégias para Tolerar Falhas em Interconexões e Aumentar o Rendimento na Produção de Redes Intra-Chip

Banca Examinadora:

Professor Doutor Ricardo Augusto da Luz Reis (UFRGS/PGMICRO)
Professor Doutor Marcelo Soares Lubaszewski (UFRGS/PGMICRO)
Professora Doutora Érika Fernandes Cota (UFRGS/PPGC)

AGRADECIMENTOS

Meus agradecimentos em geral são para todas aquelas pessoas que de alguma forma colaboraram para a construção deste trabalho, seja com idéias construtivas ou em momentos descontraídos de lazer. Cada pessoa que compartilhou alguma palavra, algum gesto, algum sorriso, algum tempo ou alguma idéia comigo durante os anos de 2009, 2010 e 2011 saberá da sua fundamental participação para a conclusão deste trabalho.

Muito obrigada a todos vocês que participaram desta jornada, tornando meus dias melhores com palavras amigas ou colaborações saudáveis.

SUMÁRIO

AGRADECIMENTOS 2
LISTA DE ABREVIATURAS 5
LISTA DE FIGURAS 7
LISTA DE TABELAS 10
1. INTRODUÇÃO 13
1.1 Estrutura do Trabalho 19
2. CONCEITOS FUNDAMENTAIS DE REDES INTRA-CHIP 20
2.1 Estrutura Básica de Redes Intra-Chip 20
2.1.1. Topologia 21
2.1.2. Roteamento 22
2.1.3. Chaveamento e Transmissão dos Dados 23
2.1.4. Controle de Fluxo e Elementos de Memória 24
2.1.5. Arbitragem 25
2.2 Definição de Métricas 26
2.3 Exemplos de Arquiteturas Encontradas na Literatura e na Indústria 26
2.4 Arquitetura Alvo: a Rede SoCIN 28
3. TRABALHOS RELACIONADOS: TOLERÂNCIA A FALHAS EM REDES INTRA-CHIP 31
3.1 Definição das Falhas 32
3.2 Trabalhos Relacionados 33
3.2.1 Técnicas Dinâmicas 34
3.2.2 Técnicas Estáticas 37
4. ABORDAGENS E ESTRATÉGIAS PROPOSTAS 42
5. ROTEAMENTO ADAPTATIVO 45

6. DIVISÃO DE DADOS NOS CANAIS DA REDE INTRA-CHIP 49
7. REMAPEAMENTO DE TAREFAS 58
5.1 Remapeamento de Redes com Núcleos Homogêneos 58
5.2 Remapeamento de Redes com Núcleos Heterogêneos 63
8. RESULTADOS DE SÍNTESE 66
8.1 Primeira Abordagem 66
8.2 Segunda Abordagem 69
9. RESULTADOS DE DESEMPENHO, ENERGIA E POTÊNCIA 72
9.1 Situações Toleradas pelo Roteamento Adaptativo na Primeira Abordagem 77
9.2 Situações Toleradas pela Divisão de Dados na Primeira Abordagem 82
9.3 Energia e Potência para a Primeira Abordagem 87
9.4 Remapeamento para a Primeira Abordagem 90
9.5 Energia e Potência para a Segunda Abordagem 92
10. ANÁLISE DE CONECTIVIDADE NA PRESENÇA DE MÚLTIPLAS FALHAS 94
11. CONCLUSÕES E TRABALHOS FUTUROS 97
11.1 Trabalhos Futuros 98
REFERÊNCIAS 101
ANEXO I: ARTIGOS PUBLICADOS 106

LISTA DE ABREVIATURAS

bop	*begin-of-packet*
BIST	*Built-In Self-Test*
CMOS	*Complementary Metal-Oxide-Semiconductor*
DVD	*Digital Video Disc*
DD	Divisão de Dados
eop	*end-of-packet*
FIFO	*First-In-First-Out*
Flit	*FLow control unIT*
GAPH	Grupo de Apoio ao Projeto de Hardware
H.264	Padrão para compressão de vídeo
MPEG4	*Moving Picture Experts Group 4*
MPSoC	*Multiprocessor System-on-Chip*
NI	*Network Interface*
NMOS	*nFET Metal Oxide Silicon*
NoC	*Network-on-Chip*
phit	*physical unit*
RA	Roteamento Adaptativo
RASoC	*Router Architecture for Systems-on-Chip*
RRADD	Roteador com Roteamento Adaptativo e Divisão de Dados
SoC	*System-on-Chip*

SoCIN	*System-on-Chip Interconnection Network*
TF	Tolerância a falhas
TMR	*Triple Modular Redundancy*
UFRGS	Universidade Federal do Rio Grande do Sul
USB	*Universal Serial Bus*
VC	*Virtual Channel*
VHDL	*VHSIC Hardware Description Language*
VHSIC	*Very High Speed Integrated Circuit*
VOPD	*Video Object Plane Decoder*
XOR	Porta Lógica "Ou Exclusivo"

LISTA DE FIGURAS

Figura 1.1: Evolução das características de um circuito integrado (GLESNER, 2010). 13
Figura 1.2**:** Organização de uma rede intra-chip de topologia grelha 3x3, com exemplos de falhas para cada componente (interconexão, núcleo, roteador e interface de rede). 16
Figura 1.3: Rendimento de acordo com a tecnologia e a distribuição de defeitos. (a) Tecnologia madura sem defeitos (caso hipotético); (b) Tecnologia madura com defeitos; (c) Tecnologia recente sem defeitos (caso hipotético) e (d) Tecnologia recente com defeitos. 17
Figura 2.1: Topologias normalmente utilizadas em redes intra-chip 22
Figura 2.2: Exemplo de uma situação em que ocorre *deadlock*. 23
Figura 2.3:Multiplexação de um canal físico em dois canais virtuais (MELO, 2006). 24
Figura 2.4: (a) Topologia grelha e torus utilizadas pela rede SoCIN e (b) formato do pacote da rede SoCIN (ZEFERINO, 2003). 29
Figura 2.5: Arquitetura básica do roteador RASoC (ZEFERINO et al., 2004). 30
Figura 3.1: MPEG4 mapeado em uma rede 4x3 torus com o exemplo de um conjunto de falhas considerado entre as interconexões. 32
Figura 3.2: Esquema de proteção contra falhas utilizando código de Hamming (FRANTZ et al., 2006). 34
Figura 3.3: Situação de falha nas interconexões que o código de Hamming não pode lidar. 35
Figura 3.4: Situação de falha nas interconexões em que nem mesmo a retransmissão dos dados de LEHTONEN et al. (2007) não pode lidar. 35
Figura 3.5:Situação de falha nas interconexões em que (BRAGA et al., 2010) não pode lidar com o uso de retransmissão e paridade. 36
Figura 3.6: Proposta de (GANGULY et al., 2009) com uso de código de Hamming e interconexões duplicadas 37
Figura 3.7: Diagrama de blocos do emissor e receptor desenvolvido por (PALESI et al., 2010). 38
Figura 3.8: Caso que a proposta de (PALESI et al., 2010) necessita evitar a interconexão defeituosa e utilizar uma função roteamento específica para atingir o destino. 38
Figura 3.9: Casos limitados de falhas que o trabalho de CONCATTO et al. (2009) não pode lidar: (a) inutilizando um roteador ou (b) não acessando um núcleo. 39
Figura 4.1: Fluxograma das etapas que compreendem o trabalho proposto. 44
Figura 5.1: Exemplo de uma situação em que o cabeçalho do pacote é alterado para mudar a rota original, a fim de evitar um caminho defeituoso. 46
Figura 5.2: Exemplo de roteamento adaptativo para um caso de uma falha entre a interconexão do roteador 3 e do roteador 6. 47
Figura 5.3: Exemplos de situações de falhas que o roteamento adaptativo não pode lidar. 48

Figura 6.1: Posicionamento dos multiplexadores para os fios de uma interconexão com 8 bits. Apenas fios bons podem ser selecionados para a transmissão dos dados. 51
Figura 6.2: Diagrama de blocos do roteador com divisão de dados. 52
Figura 6.3: Exemplo do uso da divisão de dados em uma interconexão de 8 bits com 50% de fios defeituosos. 53
Figura 6.4: Situação que exige a combinação de RA e DD para tolerar as falhas da rede. 54
Figura 6.5: Exemplos de impacto no tempo de comunicação com a utilização de DD. 55
Figura 6.6: Dois *latches* foram inseridos após a redução do bloco DD para minimizar o impacto na comunicação. 57
Figura 6.7: Diagrama de dados em uma interconexão que utiliza DD com *latches*..... 57
Figura 7.1: Grafo de comunicação para um caso hipotético. 59
Figura 7.2: Mapeamento escolhido para o grafo da comunicação anterior..... 59
Figura 7.3: Tempo de comunicação para cada núcleo da rede no mapeamento escolhido. 60
Figura 7.4: Situação crítica em que é detectada uma falha no *CR_link* do núcleo A. 60
Figura 7.5: Possibilidades de mapeamento adotadas com o espelhamento dos núcleos..... 61
Figura 7.6: Tempo de comunicação para cada núcleo quando mapeado no *CR_link* defeituoso de acordo com a figura 7.5. 62
Figura 7.7: Exemplos de replicação que permitem o remapeamento utilizando a topologia torus. 64
Figura 9.1: Padrão de comunicação do VOPD. 73
Figura 9.2: Padrão de comunicação do MPEG4. 73
Figura 9.3: Padrão de comunicação do H.264. 74
Figura 9.4: Padrões de tráfego sintético: complementar e borboleta. 74
Figura 9.5: Nomenclatura das interconexões em uma rede 4x3. 75
Figura 9.6: Quantidade de conexões afetadas pelo roteamento adaptativo para o VOPD..... 78
Figura 9.7: Quantidade de conexões afetadas pelo roteamento adaptativo para o MPEG4. 79
Figura 9.8: Quantidade de conexões afetadas pelo roteamento adaptativo para o H.264. 80
Figura 9.9: Impacto na soma dos tempos de comunicação para 3 situações. 83
Figura 9.10: Número de núcleos que são afetados pela falha no *RC_link* entre o núcleo e seu respectivo roteador. 84
Figura 9.11: Impacto no tempo final de execução da aplicação de acordo com a localização da falha. 84
Figura 9.12: Impacto na soma de todos os tempos de cada comunicação de acordo com a localização da falha. 85
Figura 9.13: Número de núcleos que são afetados pela falha no *RC_link* entre o núcleo e o seu respectivo roteador. 86
Figura 9.14: Impacto na soma dos tempos de cada comunicação de acordo com a localização da falha. 87
Figura 9.15: Comparativo de energia entre as propostas abordadas após a utilização do remapeamento. 92
Figura 9.16: Comparativo de energia entre cada abordagem analisada para 8 bits. 92
Figura 9.17: Comparativo de energia entre cada abordagem analisada para 32 bits. 93
Figura 9.18: Impacto em % de energia para cada estratégia analisada. 93
Figura 10.1: Conectividade da rede de acordo com o número de falhas nas interconexões. 95
Figure 10.2: Análise da conectividade para as estratégias analisadas neste trabalho. 96

Figura 10.3: Capacidade de cada estratégia para lidar com múltiplos fios defeituosos. 96
Figura 11.1: Atraso relativo entre os fios e a lógica de um circuito de acordo com a tecnologia utilizada (ITRS, 2009). 99

LISTA DE TABELAS

Tabela 2.1: Comparação entre redes Intra-Chip. 27
Tabela 3.1: Comparação entre os trabalhos relacionados. 41
Tabela 7.1: Tempo de comunicação total para cada situação de mapeamento. 62
Tabela 8.1: Comparações entre os resultados de síntese. 68
Tabela 8.2: Resultados estimados de potência para todos os fios das redes 4x3 e 4x4. 68
Tabela 8.3: Resultados de síntese para a nova abordagem do roteador RRADD. 70
Tabela 8.4: Resultados de potência em cada fio de acordo com cada benchmark utilizado. 71
Tabela 9.1: Número de interconexões utilizadas em uma rede 4x3 e 4x4. 76
Tabela 9.2: Comparações entre alguns resultados de RA para os 3 benchmarks analisados. ... 80
Tabela 9.3: Comparações entre alguns resultados de RA para os 2 tráfegos sintéticos analisados. 82
Tabela 9.4: Tempo médio total de comunicação para cada caso analisado. 88
Tabela 9.5: Potência para cada rede analisada. 89
Tabela 9.6: Resultados de energia, considerando o comprimento médio dos fios de 1 mm. 89
Tabela 9.7: Resultados do remapeamento para alguns casos analisados. 91
Tabela A: Trabalhos publicados durante o mestrado acadêmico. 107

Combination of Strategies to Tolerate Faults in the Interconnections and to Increase the Yield in the Manufacture of Networks-on-Chip

ABSTRACT

A Network-on-Chip (NoC) can offer better scalability and performance than a traditional bus, and therefore it has been used as an alternative communication architecture inside of a complex System-on-Chip. The use of fault tolerance structures in NoC is growing, due to the fact that it is almost impossible to manufacture integrated circuits without any defect in nanometer technologies. Consequently, the use of fault tolerance methods is crucial to allow that circuits with some amount of defects still reach the market, increasing yield and the lifetime of a chip, besides ensuring the correct functionality of the device. Based on previous test and diagnosis results, the NoC can have embedded fault-tolerant solutions that can provide the correct communication in the network.

A strategy to handle multiple defects in the NoC interconnections with low impact on the communication delay and energy is presented in this thesis. The fault-tolerant method can guarantee the functionally of the NoC with multiple defects in any interconnection, and with multiple faulty interconnections. The proposed techniques use information from testing to adapt the routing and the packet, which allows configuring fault-tolerant features along the NoC interconnections. A remapping strategy can be associated to minimize the impact of some faults in the application.

Results for the combination of three different techniques in the NoC show that the communication delay can have minimal impact when compared to a fault-free system. Comparisons have shown that our proposal can provide a better fault tolerance against permanent faults than Hamming code in terms of energy and performance impact. We show that the proposed strategy has a minimized impact in performance and power while a traditional fault-tolerant solution like Hamming code has a significant impact.

Keywords - NoCs, Fault Tolerance, Interconnections, Adaptive Routing, Data Splitting, Mapping, Yield, Microelectronics.

RESUMO

Uma rede intra-chip pode oferecer melhor desempenho e escalabilidade do que um barramento tradicional, e, portanto, ela tem sido utilizada como uma arquitetura alternativa de comunicação dentro de um complexo sistema intra-chip. O uso de estruturas tolerantes a falhas em rede intra-chip está crescendo, devido ao fato de ser quase impossível produzir circuitos integrados sem qualquer defeito em tecnologias nanométricas. Consequentemente, o uso de tolerância a falhas é crucial para permitir que circuitos com alguma quantidade de defeitos ainda alcancem o mercado, incrementando o rendimento e o tempo de vida de um chip, além de garantir a correta funcionalidade do dispositivo. Com base nos resultados prévios de teste e diagnóstico, a rede intra-chip pode ter soluções embarcadas tolerante a falhas que podem proporcionar a correta comunicação na rede.

Uma estratégia para manipular múltiplos defeitos nas interconexões da rede intra-chip com baixo impacto no atraso da comunicação e em energia é apresentada nesta dissertação. O método tolerante a falhas pode garantir a funcionalidade da rede com múltiplos defeitos em qualquer interconexão, e com múltiplas interconexões defeituosas. As técnicas propostas usam a informação do teste para adaptar o roteamento e o pacote de dados permitindo configurar as características de tolerância a falhas entre as interconexões da rede intra-chip. Uma estratégia de remapeamento pode ser associada para minimizar o impacto de algumas falhas na aplicação.

Resultados para a combinação de três diferentes técnicas na rede intra-chip mostram que o atraso na comunicação pode ter impacto mínimo quando comparado com o sistema livre de falhas. Comparações tem mostrado que nossa proposta pode proporcionar uma melhor tolerância a falhas contra falhas permanentes do que Hamming. Nós mostramos que a estratégia proposta tem um impacto reduzido no desempenho e na potência enquanto que uma solução tradicional como código de Hamming tem um impacto significativo.

Palavras-chave – Redes intra-chip, tolerância a falhas, interconexões, roteamento adaptativo, divisão de dados, mapeamento, rendimento, microeletrônica.

1. INTRODUÇÃO

O desenvolvimento da tecnologia de fabricação e o avanço da microeletrônica desencadearam um crescente número de equipamentos eletrônicos nas últimas décadas (CARRO, 2001). Essa diversidade de equipamentos atingindo o consumidor a uma velocidade cada vez maior só é possível porque foram dedicados muitos anos de pesquisa e trabalho no desenvolvimento de hardware. Certa vez, em 1965, Gordon Moore fez a uma afirmação de que a cada 18 meses o número de transistores por área em um determinado circuito iria dobrar. Desde então, as principais empresas responsáveis pelo desenvolvimento de componentes eletrônicos seguem rigorosamente a afirmação que ficou conhecida como a "*Lei de Moore*" (MOORE, 1965). Na época, Gordon Moore trabalhava na *Fairchild Corporation*, e tornou-se mais tarde o co-fundador da *Intel*, empresa com alto domínio sobre as tecnologias mais recentes para o desenvolvimento de transistores em geral, mais especificamente para compor processadores. A Lei de Moore vem sendo aplicada em muitos segmentos na área de circuitos integrados, integrando cada vez mais os componentes dentro de um único chip, aumentando a complexidade e a performance. A figura 1.1 mostra a evolução histórica do tamanho e da quantidade de transistores em circuitos integrados, desde 1960 até os dias de hoje, confirmando a Lei de Moore.

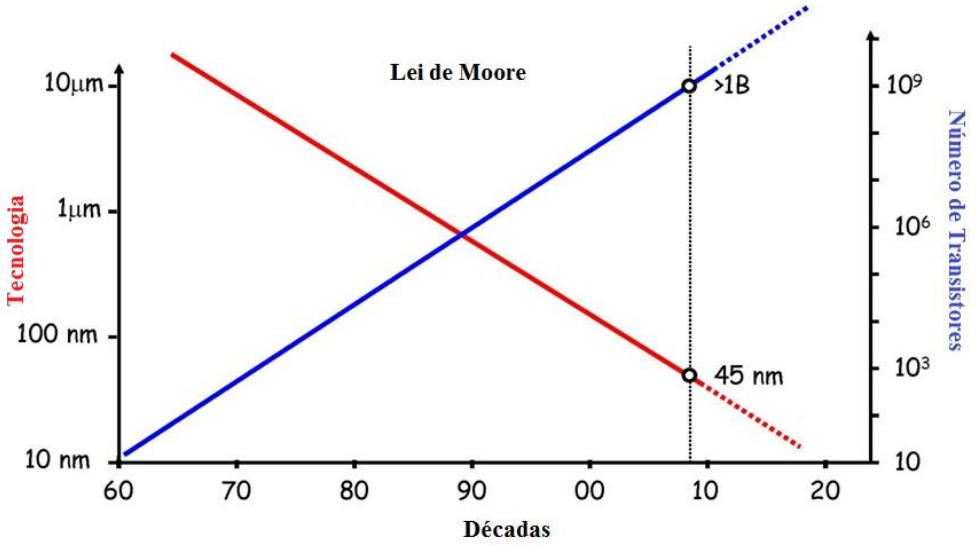


Figura 1.1: Evolução das características de um circuito integrado (GLESNER, 2010).

Com o avanço da tecnologia, é muito comum integrar diversos componentes dentro de um único circuito através do uso de *Systems-on-Chip* (SoCs). Com base nos SoCs, surgiram os MPSoCs (*Multiprocessor System-on-Chip*), a fim de suportarem a crescente complexidade dos sistemas embarcados, uma vez que as arquiteturas são adaptadas a uma classe de funcionalidades requeridas, aliadas juntamente com a fácil flexibilidade de programação que estes dispositivos permitem. Desta forma, surge uma crescente necessidade de comunicação entre os componentes de um SoC ou MPSoC, e consequentemente surge um amplo espaço de projeto para inserir neste contexto as redes intra-chip no mercado de componentes eletrônicos. Desde o ano de 2000, quando foram publicados alguns dos primeiros trabalhos significativos sobre redes intra-chip (GUERRIER E GREINER, 2000) (LANGEN, 2000), o interesse pelo assunto tem sido tema de sessões especiais em importantes conferências internacionais. Redes intra-chip são uma alternativa ao uso clássico de barramentos, e proporcionam a comunicação de diferentes componentes dentro de um circuito, solucionando a baixa escalabilidade e a crescente complexidade envolvidas na utilização de barramentos.

O problema do barramento surgiu quando muitos componentes foram integrados em um mesmo SoC. Este problema é proveniente da comunicação simultânea entre muitos componentes, o que significa o compartilhamento do barramento por onde os dados trafegam, ou múltiplos barramentos para atender cada uma das comunicações ao mesmo tempo. Se a primeira opção é escolhida, o atraso (ou latência) da rede pode incrementar, já que apenas uma comunicação utilizará o barramento por vez enquanto possivelmente múltiplas fontes necessitam se comunicar ao mesmo tempo (NICOPOLOUS et al., 2009). Se múltiplos barramentos são utilizados, então a complexidade no controle dos barramentos aumenta e o posicionamento se torna dependente de cada aplicação, de acordo com os destinos da comunicação (BENINI E DE MICHELI, 2002). Para solucionar este problema, as redes intra-chip são projetadas de modo muito semelhante às tradicionais redes de computadores que conhecemos, proporcionando uma rede genérica e modular que pode ser utilizada em diferentes aplicações, integrando diversos componentes e provendo um alto paralelismo na comunicação dentro de um único e complexo SoC. Desta forma, as redes intra-chip proporcionam a comunicação entre múltiplos componentes de uma forma mais eficiente.

As redes intra-chip são também conhecidas amplamente na área acadêmica por *Networks-on-Chip*, ou pela sigla NoCs (GUERRIER E GREINER, 2000) (LANGEN, 2000) (BENINI E DE MICHELI, 2002), e elas são basicamente utilizadas para melhorar os comprometimentos da rede de modo a trazer diversos benefícios e favorecendo a ampla comunicação entre todos os componentes de um chip. Redes intra-chip podem agregar técnicas que visam melhorar muitos compromissos, como, por exemplo, relações que maximizam a eficiência da comunicação com a mínima energia possível e máximo desempenho (JINGCAO et al., 2006), além de técnicas para tolerar diferentes tipos de falhas que podem ser também incluídas no projeto para a proteção do sistema (MURALI, 2009) (NICOPOLOUS et al., 2009).

O projeto de redes intra-chip tipicamente almeja uma aplicação específica ou uma limitada classe de aplicações quando se pretende garantir alguns compromissos. Contudo, pesquisas recentes mostram que existem razões para projetar-se uma rede adaptativa a fim de sustentar computações de propósito geral ou a fim de prover confiabilidade (NICOPOLOUS et al., 2006) (STENSGAARD E SPARSO, 2008) (LAN et al., 2009) (CONCATTO et al., 2009). Desta forma, o uso de redes intra-chip torna-se amplo devido à facilidade de reconfiguração e adaptabilidade que elas podem proporcionar de acordo com o projeto executado, e a incorporação de soluções para garantir o funcionamento adequado tem se tornado essencial. Como as redes intra-chip possuem uma estrutura bastante similar às tradicionais redes de computadores, a topologia também pode tornar-se um parâmetro de escolha bastante crítico para prover a comunicação ideal de acordo com os requisitos do sistema. Além disso, é muito comum determinados circuitos serem reutilizados em diferentes aplicações, ou até mesmo existir diferentes padrões de tráfego no comportamento de uma mesma aplicação, como é o caso do vídeo game Xbox, que pode carregar e salvar um jogo, além de ler fotos por uma porta USB dentre outras opções (ANDREWS E BAKER, 2006).

Redes intra-chip são basicamente compostas por roteadores, canais de comunicação conhecidos por links, interconexões ou canais, e interfaces de rede (referenciadas por NI, sigla que corresponde a *Network Interface*), conectando todos elementos de processamento ou núcleos (ou ainda *cores*, em inglês) que precisam se comunicar, proporcionando amplo paralelismo na comunicação, como mostra a figura 1.2. Normalmente as interfaces de rede podem ser omitidas a fim de simplificar a apresentação da rede. Porém, a integração de muitos componentes em um único chip trouxe severas preocupações para os projetistas de circuitos integrados com relação à confiabilidade e ao rendimento dos chips produzidos, uma vez que diversos tipos de falhas podem acontecer, como ilustrado também na figura 1.2. Falhas podem ocorrer em diversos pontos: elementos de processamento (núcleos), interconexões, roteadores ou interfaces de rede. Consequentemente, técnicas de tolerância a falhas estão sendo amplamente estudadas e embarcadas nos circuitos para garantir um rendimento mínimo e uma certa confiabilidade aos sistemas, e as falhas podem ser estudadas de acordo com o seu tipo e/ou localização.

Porém, um problema muito comum surgiu com a evolução da tecnologia, e está sendo estudado amplamente por diversos grupos de pesquisa nos dias de hoje: o surgimento de falhas devido a diferentes circunstâncias. Com a miniaturização dos transistores, a manufatura dos circuitos tornou-se muito delicada, sujeita a defeitos permanentes de fabricação oriundos de partículas de pó, e de imprecisão ou má calibração das medidas e dosagens responsáveis pela fabricação do transistor, por exemplo. Além disso, existem fontes de falhas transientes como interferências, pulsos eletromagnéticos, radiação e partículas provenientes do espaço e do sol que podem atingir o circuito e interferir na resposta final alterando o resultado esperado. Estes problemas tornaram-se mais comuns com o avanço da tecnologia, pois normalmente em transistores produzidos com tecnologias maduras (transistores mais antigos), as falhas

não eram suficientes para causar danos alterando a condução. Para solucionar o problema da presença de falhas em um circuito, mecanismos de proteção são utilizados para garantir a correta funcionalidade do equipamento, atribuindo a devida confiabilidade ao dispositivo. Assim, é necessário proteger constantemente os circuitos desenvolvidos em tecnologias com pouca maturidade (tecnologias recentes), para garantir a correta funcionalidade dos dispositivos, mesmo que a maioria das técnicas conhecidas tenham um alto custo para proporcionar a devida proteção ao circuito. Além disso, as falhas podem se manifestar combinadas simultaneamente, caracterizando situações de falhas múltiplas nos dispositivos, exigindo algum tipo mais robusto de tolerância a falhas para lidar com tais circunstâncias. Então, falhas em tecnologias com escala nanométrica tornaram-se inevitáveis, e por isso a utilização de recursos para lidar com situações de falhas tornou-se mandatória (FURBER, 2006), (HOSSEINABADY e NUNEZ-YANEZ, 2008).

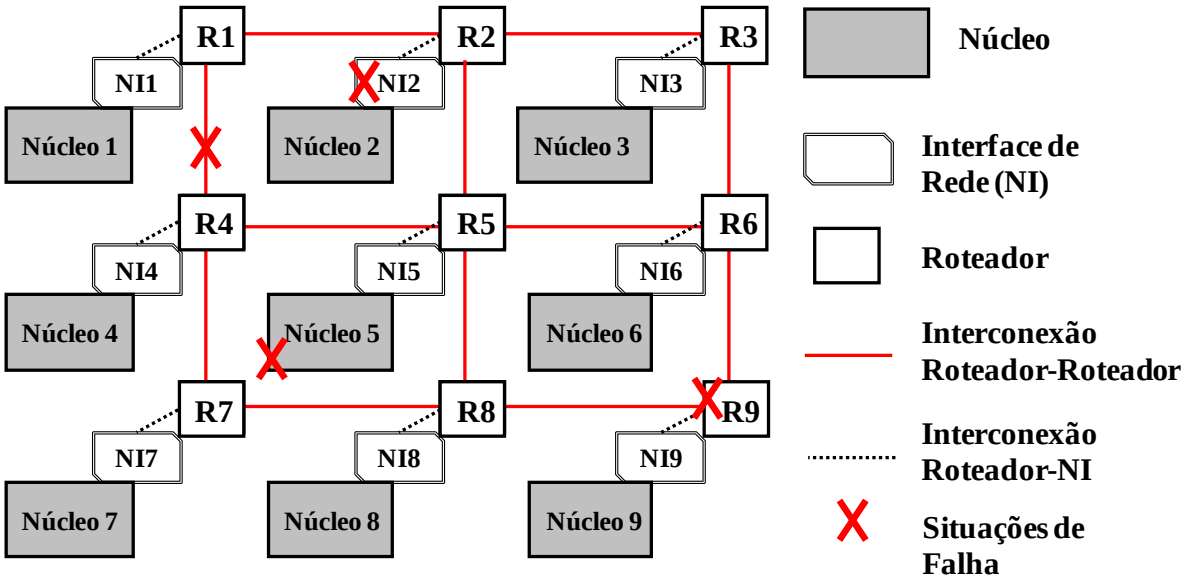
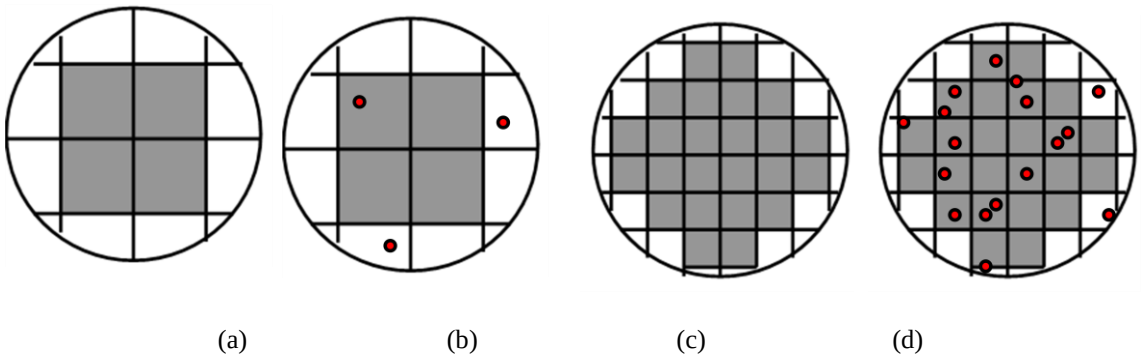


Figura 1.2**:** Organização de uma rede intra-chip de topologia grelha 3x3, com exemplos de falhas para cada componente (interconexão, núcleo, roteador e interface de rede).

Em 2006, de acordo com projeções realizadas pela Intel, Steve Furber deixou claro a seguinte afirmação para a próxima década, considerando o significativo avanço da tecnologia no processo de manufatura: "chips com 100 bilhões de transistores terão em torno de 20% de transistores produzidos com defeitos já no processo de manufatura, e outros 10% irão falhar no primeiro ano de operação" (FURBER, 2006). Sendo assim, não restam dúvidas de que técnicas para proteger os circuitos desenvolvidos com tecnologias novas serão fundamentais para garantir o correto funcionamento dos circuitos projetados atualmente, pelo menos até o processo de manufatura atingir uma maturidade considerável, embora para transistores de canais curto a suscetibilidade a falhas estará sempre presente (COLINGE, 2008).

Outro fator que contribui consideravelmente para o desenvolvimento da tecnologia é o rendimento (RABAEY et al., 2003). Quando circuitos são projetados em uma pastilha de silício utilizando tecnologias maduras e amplamente consolidadas, significa que é possível obter um bom rendimento por lâmina de silício, pois apesar de cada circuito ter uma área consideravelmente grande (com poucos circuitos produzidos dentro da pastilha de silício) a maioria dos circuitos não terá falhas devido ao amadurecimento do processo de manufatura utilizado. Quando a tecnologia tornou-se mais aprimorada (ou seja, os transistores diminuíram), mais chips puderam ser obtidos considerando a mesma área da lâmina de silício, embora o rendimento não possa ser favorecido na mesma proporção como mostra a figura 1.3, uma vez que o processo de manufatura está mais sujeito a ocorrência de falhas, ocasionando um rendimento menor. Cada quadrado na figura 1.3 representa um circuito projetado e cada pontinho que atinge um circuito válido representa uma falha que inutiliza ou prejudica o circuito em questão. As extremidades (ou bordas) são normalmente descartadas devido a suas formas irregulares.



(a) (b) (c) (d)

Figura 1.3: Rendimento de acordo com a tecnologia e a distribuição de defeitos. (a) Tecnologia madura sem defeitos (caso hipotético); (b) Tecnologia madura com defeitos; (c) Tecnologia recente sem defeitos (caso hipotético) e (d) Tecnologia recente com defeitos.

Outra peculiaridade que as tecnologias recentes apresentam é a presença de múltiplas falhas em um mesmo circuito produzido, necessitando então de mecanismos para lidar não apenas com situações de falha única, mas com múltiplas ocorrências de falhas dentro de um mesmo circuito, acarretando na aplicação de técnicas com maior complexidade para garantir a proteção adequada.

Existem muitos trabalhos na literatura que protegem diferentes partes das redes intra-chip contra diferentes tipos de falhas. Alguns exemplos de trabalhos que podem ser encontrados na literatura são: FRANTZ et al. (2006), SCHONWALD et al. (2007), LEHTONEN et al. (2007), KOIBUCHI et al. (2008), GANGULY et al. (2009), CONCATTO et al. (2009), TORNERO et al. (2009), CHOUDHURY et al. (2009), BRAGA et al. (2010), PALESI et al. (2010) e KAKOEE et al. (2011). As falhas normalmente podem ser classificadas em falhas permanentes, transientes ou intermitentes e a detecção das falhas pode ser estática ou dinâmica. Técnicas estáticas requerem a prévia execução de testes no circuito em questão, permitindo a alocação (ou

não) de recursos conforme a necessidade do circuito. Técnicas dinâmicas são sempre executadas durante toda a vida útil do circuito, e podem ser redundantes enquanto a presença de uma falha não é manifestada. Normalmente, técnicas dinâmicas caracterizam um impacto constante e estão sempre presentes nos circuitos em que são empregadas. Algumas técnicas dinâmicas permitem corrigir além de apenas detectar as falhas, embora tenham uma baixa eficiência (normalmente poucos bits de uma informação podem ser corrigidos).

Para evitar um novo projeto e permitir a reutilização das redes intra-chip em diferentes circunstâncias, o uso de mecanismos para tolerar falhas pode ser muito útil, de modo que o rendimento possa ser incrementado. Desta forma, esta trabalho propõe utilizar a combinação de técnicas para prover tolerância a falhas em tempo de manufatura, a fim de atuar contra múltiplas situações de falhas nas interconexões da rede intra-chip. As técnicas foram desenvolvidas de modo a permitir a escolha da utilização, a fim de configurar a rede para proporcionar confiabilidade com o mínimo impacto desejado em tempo e energia na execução das tarefas, de acordo com a localização das falhas.

Essa dissertação foca em falhas do tipo permanente e intermitentes nas interconexões, e tem por principal objetivo aumentar o rendimento da rede intra-chip, fazendo com que interconexões defeituosas possam ainda ser utilizadas sem grandes prejuízos, como será mostrado ao longo deste trabalho. Falhas permanentes podem ser modeladas como falhas de curto circuito entre fios de um mesmo canal ou canais distintos, por exemplo. E as falhas intermitentes são falhas de crosstalk seguindo o modelo MAF (CUVIELLO et al., 1999).

Então, atualmente, o uso de técnicas de tolerância a falhas (TF) pode ser a única alternativa para manter o rendimento na fabricação de SOCs em geral. Porém a utilização de técnicas contra falhas múltiplas pode trazer um alto custo em área, desempenho e potência para as redes intra-chip em geral, como acontece com a maioria dos trabalhos propostos para lidar com esse tipo de situação. O uso de técnicas adaptativas neste trabalho justifica-se porque visa reduzir estes custos, aproveitando para aumentar o rendimento devido a configuração de TF somente em regiões defeituosas, que realmente precisam utilizar as técnicas de proteção. Porém, técnicas adaptativas precisam agregar um mecanismo que detecte as falhas, seja em tempo de execução ou com base no resultado prévio dos testes realizados. Assim, o objetivo desta dissertação também consiste em mostrar que é possível minimizar os custos em geral por utilizar a informação prévia do teste.

As técnicas apresentadas neste trabalho podem ser resumidas basicamente na combinação de três estratégias: roteamento adaptativo, divisão de dados e remapeamento, que foram escolhidas justamente pela simplicidade que cada uma delas apresenta, tanto conceitualmente quanto em termos de implementação. Cada uma das abordagens será apresentada em detalhes neste trabalho, e a escolha de cada técnica

também será explorada, levando em conta principalmente o custo do hardware extra e o nível desejado de proteção.

1.1 Estrutura do Trabalho

No capítulo 2 são apresentadas as principais características, parâmetros e conceitos referentes a uma rede intra-chip, e algumas redes amplamente conhecidas na literatura serão abordadas e comparadas. Também são apresentadas no capítulo 2 as características da arquitetura alvo utilizada neste trabalho.

O principal objetivo deste trabalho é combinar estratégias para prover a confiabilidade das interconexões da rede, embora outras partes defeituosas da rede possam ser protegidas pela combinação das técnicas propostas, com mínimo impacto em tempo de execução, potência e energia, a fim de garantir um melhor rendimento na produção. Por isso, no capítulo 3 serão apresentados os trabalhos relacionados, e alguns conceitos básicos referentes a tolerância a falhas. Diferentes níveis de proteção em diversas regiões da rede intra-chip, abordados por outros trabalhos, também serão explorados no capítulo 3.

No capítulo 4 as estratégias adotadas neste trabalho são descritas brevemente, e nos capítulos 5, 6 e 7 cada estratégia adotada neste trabalho é abordada com mais detalhes, mostrando em profundidade as técnicas utilizadas para prover a tolerância a falhas, que são respectivamente: o roteamento adaptativo, a divisão dos dados e o remapeamento das tarefas. No capítulo 8 são apresentados os resultados de síntese para cada estratégia. No capítulo 9, resultados de desempenho, energia e potência serão discutidos, e a conectividade proporcionada pela combinação das técnicas na rede, mesmo na presença de falhas, é apresentada no capítulo 10, para uma análise considerando situações com múltiplas falhas.

Para finalizar, as conclusões e as ideias de trabalhos futuros encontram-se no capítulo 11. Algumas considerações finais foram também anexadas ao trabalho, a fim de relacionar os trabalhos desenvolvidos ao longo do mestrado acadêmico.

2. CONCEITOS FUNDAMENTAIS DE REDES INTRA-CHIP

BENINI e DE MICHELI mostraram em 2002 que o desempenho e a potência de um chip são muito prejudicados pelas interconexões e barramentos tradicionais. Por isso, as redes intra-chip estão se tornando de fato uma solução para prover comunicação entre complexos *Systems-on-Chip*, integrando diferentes núcleos. Com o objetivo de proporcionar benefícios aos projetistas, então as redes intra-chip estão tornando-se amplamente adotadas por diferentes empresas para compor os equipamentos eletrônicos, como é caso da *STMicroelectronics*, e mais recentemente, da Intel.

Neste capítulo, serão apresentados os conceitos básicos e as características da arquitetura de uma rede intra-chip, e também serão abordadas e comparadas algumas redes amplamente conhecidas na literatura.

2.1 Estrutura Básica de Redes Intra-Chip

A arquitetura de uma rede intra-chip pode ser projetada de modo genérico, mas para atender certas restrições ela fundamentalmente necessita de um projeto personalizado, a fim de satisfazer as melhores condições de operação para cada aplicação em questão. Redes intra-chip podem normalmente atender restrições de potência, área e desempenho. Porém, existem outras restrições que podem ser impostas ao projeto, a fim de obterem-se melhores resultados como, por exemplo, baixa latência, altas taxas de comunicação e alta confiabilidade. Por isso é que existem diferentes projetos de redes intra-chip na literatura, pois cada projeto visa um conjunto de restrições diferentes. Mas todas as redes, além das próprias peculiaridades, possuem características essenciais em comum que podem ser citadas, como é o caso da topologia da rede, do tipo de roteamento, do tipo de chaveamento, da transmissão dos dados, do controle do fluxo, do armazenamento dos dados, e também da arbitragem (ZEFERINO, 2003). Muitas arquiteturas permitem a parametrização de alguns destes parâmetros, e cada um deles é abordado individualmente neste capítulo.

2.1.1. Topologia

A topologia é uma característica fundamental para a composição das redes intra-chip, pois a estrutura dela pode impactar diretamente no tempo de comunicação e é de acordo com ela que o tipo de roteamento a ser utilizado é escolhido. As principais topologias de rede intra-chip para 2 dimensões são classificadas como consta a seguir, e são normalmente conhecidas pela nomenclatura em inglês:

- *Mesh, Grid ou Grelha*: é a topologia mais utilizada devido ao alto grau de regularidade. Basicamente, é composta por linhas e colunas regulares de roteadores, com comunicação através de interconexões verticais e horizontais;
- *Torus*: é composta pela topologia grelha, exceto que as extremidades se conectam diretamente através de uma malha de realimentação. Essa topologia permite uma comunicação mais rápida entre os roteadores que se encontram nas extremidades opostas da rede, e também possibilita caminhos diferentes para atingir um mesmo destino de acordo com as possibilidades de configuração do roteamento;
- *Ring ou Anel*: possui formato circular, e como consequência os roteadores estão todos conectados em série. O atraso pode ser considerável se a rede for muito grande, embora a confiabilidade de um caminho único para os pacotes fortaleça a garantia de entrega;
- *Star ou Estrela*: utiliza um roteador no centro que se conecta a todos os outros roteadores da rede. Com esta topologia, o caminho entre uma fonte e um destino é na maioria das vezes pequeno, com latência reduzida, embora o roteador central permaneça constantemente congestionado;
- *Tree ou Árvore*: topologia organizada através de ramificações. Uma topologia árvore pode ser do tipo binária, a qual cada ramo tem apenas duas subdivisões, do tipo *fat-tree* (árvore gorda), em que a largura do canal cresce de acordo com a altura da árvore, ou irregular com um variado número de ramos e subdivisões. Normalmente essa topologia é utilizada quando os ramos próximos têm uma comunicação bastante intensa, permitindo que a distância seja pequena entre eles;
- *Butterfly ou Borboleta*: em alguns casos pode ser semelhante à topologia árvore gorda, mas normalmente é organizada com interconexões diretas e cruzadas de forma alternada. Existe a ausência de comunicação direta na vertical ou na horizontal, dependendo do ângulo em que se observa o grafo de comunicação. Em definições mais formais que utilizam coordenadas binárias, os bits mais significativos são trocados com os bits menos significativos, e a comunicação acontece entre cada um desses pares ordenados;
- *Hybrid ou Híbrida*: contém mais de uma das topologias apresentadas anteriormente. Se cada parte da rede tem um comportamento diferente, a mistura entre as topologia pode ser muito útil quando o padrão de comunicação é variado. É necessário um algoritmo de roteamento especial capaz de lidar com as diferentes topologias;
- *Irregular*: pode ter qualquer formato diferente das topologias citadas.

A figura 2.1 ilustra os casos apresentados para 2 dimensões. Existem ainda topologias para 3 dimensões que não serão abordadas neste trabalho.

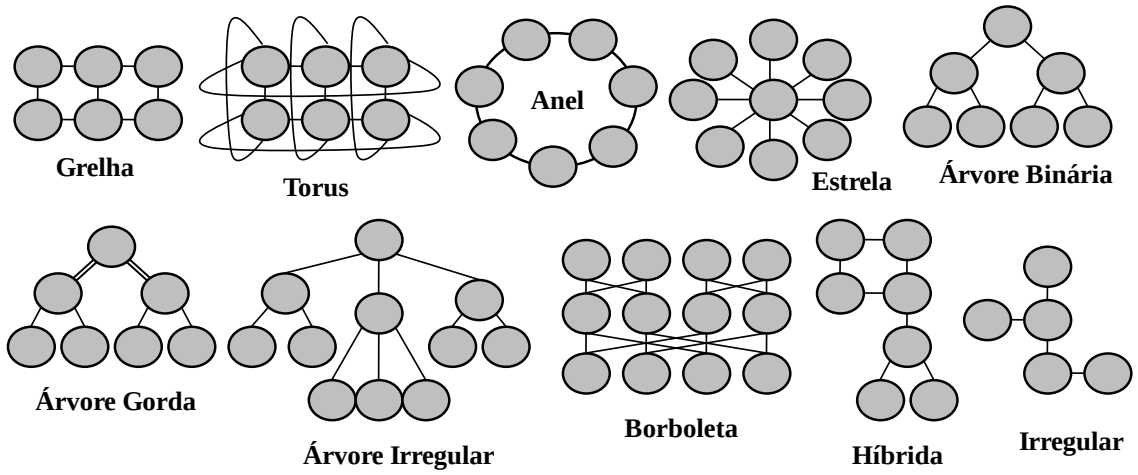


Figura 2.1: Topologias normalmente utilizadas em redes intra-chip.

2.1.2. Roteamento

O algoritmo de roteamento pode normalmente ser classificado como determinístico ou adaptativo (ZEFERINO, 2003). O algoritmo de roteamento determinístico utiliza sempre um mesmo caminho entre a fonte e o destino de uma comunicação, e ele é baseado em um algoritmo estático que decide qual o caminho a ser tomado em função do roteador fonte e do roteador alvo. O algoritmo de roteamento adaptativo é definido em função do tráfego da rede, e pode escolher em tempo de execução qual o melhor caminho a ser tomado para atingir o roteador alvo, evitando, por exemplo, congestionamentos e caminhos defeituosos.

Alguns exemplos de algoritmos de roteamento amplamente utilizados na literatura são citados a seguir:

- *XY*: normalmente esse algoritmo de roteamento é aplicado em redes com topologia grelha ou torus. Os dados percorrem todo o caminho possível em uma única direção, e somente quando não é possível atingir o alvo pela direção utilizada é que a direção é alterada. Isso significa que os dados percorrem toda a direção X, por exemplo, atingindo a coluna onde está o alvo da comunicação, e somente após alcançar esta coluna é que os dados mudam para a direção Y em busca do alvo da comunicação (MELO, 2005) (ZEFERINO E SUSIN, 2003);
- *Odd-Even ou Par-Ímpar:* é também aplicado em redes com topologia grelha ou torus. Basicamente, ele permite aos dados trafegarem por uma direção qualquer, impedindo que eles retornem pela mesma direção. Desta forma, ele é muito semelhante ao algoritmo XY, mas não necessita iniciar a comunicação sempre por uma direção pré-determinada, podendo alterar os eixos sem qualquer prejuízo (CHIU, 2000) (ZHANG et al., 2009);

- *Tabelas:* normalmente é utilizado em topologias irregulares e tem uma alta penalidade em área e potência, já que tabelas correspondem a elementos de memória configurados em arranjos (BERTOZZI E BENINI, 2004)(PALERMO et al, 2007) (GOOSSENS E HANSON, 2010).

A comunicação é realizada com sucesso quando os dados enviados atingem o alvo corretamente. Porém, existem três situações que podem impedir uma comunicação de forma adequada: *starvation*, *livelock* e *deadlock* (ZEFERINO, 2003).

A situação de *starvation* acontece quando dois canais de entrada requisitam um mesmo canal de saída, e por alguma razão a arbitragem opta por priorizar um único canal, deixando o outro impossibilitado de realizar a comunicação por um longo período de tempo. Quando os dados trafegam permanentemente na rede devido a impossibilidade de atingir o alvo (já que os recursos necessários para isso estão ocupados), a situação recebe o nome de *livelock.* Quando existe uma dependência cíclica na rede impossibilitando a comunicação porque nenhuma das partes interessadas consegue obter o conjunto de recursos necessários, ocorre o que se denomina de *deadlock*, como mostra a figura 2.2.

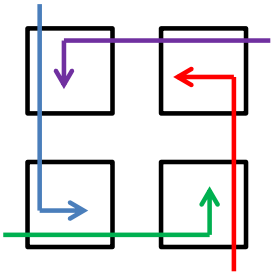


Figura 2.2: Exemplo de uma situação em que ocorre *deadlock*.

2.1.3. Chaveamento e Transmissão dos Dados

Para transmitir dados, as redes intra-chip os distribuem em pacotes de um determinado tamanho (por exemplo: 8, 16, 32, 64 ou 128 bits), que normalmente são subdivididos em unidades menores chamadas de flits (sigla para *FLow control unIT*, que significa unidade de controle de fluxo). Sendo assim, um pacote pode ser composto por um determinado número de flits, que pode ser um parâmetro variável de acordo com a implementação. Então, pacotes que saem de uma determinada fonte com um destino em comum podem atingi-lo utilizando dois tipos de chaveamento, conhecidos por *circuit switching* e *packet switching*, que podem ser aplicados para redes em geral (BERTOZZI E BENINI, 2004) (STALLINGS, 2007).

O chaveamento por circuito aloca um único caminho para cada comunicação, sendo o caminho do primeiro pacote transmitido exatamente o mesmo para todos os outros pacotes subseqüentes, tornando-o semelhante a um circuito fechado para cada pacote transmitido. Este caminho somente será desfeito quando o último pacote for

transmitido, desalocando o canal de comunicação. Enquanto isso não acontecer, torna-se inviável que outros canais utilizem ou compartilhem o mesmo caminho de comunicação.

Quando apenas o chaveamento por pacotes é utilizado, significa que cada pacote pode ser subdivido em flits, e assim o pacote inteiro é enviado em partes. Esse chaveamento também pode ser conhecido por *wormhole*, já que o pacote é divido e todos os flits de dados (*payload*) seguem o flit de cabeçalho (o primeiro flit transmitido, que contém o endereço de destino).

Além dos dois modos de chaveamento normalmente utilizados, existe ainda o que se chama de canal virtual, que influencia significativamente na transmissão dos dados. Um canal virtual (conhecido por VC, de *Virtual Channel*) utiliza múltiplos buffers de armazenamento para cada canal e permite aumentar os recursos de alocação para cada pacote transmitido (MELO, 2006). O uso de um canal virtual ocorre com a multiplexação dos dados para prover qualidade de serviço (QoS). Assim, se um canal é requisitado por múltiplas fontes, o uso dele torna-se intercalado, e o armazenamento dos dados de cada fonte acontece em buffers separados próprios para isso, como mostra a figura 2.3. O sinal denominado "E" representa o controle utilizado para determinar qual canal virtual tem acesso ao canal físico em cada instante.

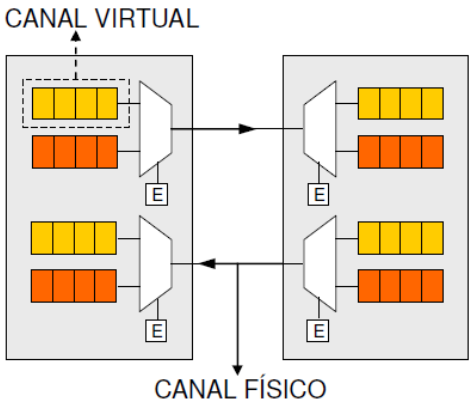


Figura 2.3:Multiplexação de um canal físico em dois canais virtuais (MELO, 2006).

2.1.4. Controle de Fluxo e Elementos de Memória

Um roteador deve ser capaz de armazenar todos os pacotes destinados as suas saídas quando elas já estão sendo utilizadas por outros pacotes e, então, realizar o controle de fluxo de modo que nenhum dado seja perdido nos seus canais de entrada, ou pelo menos reduzir o número de pacotes descartados. Isso exige a implementação de algum esquema de memorização para a manutenção dos pacotes bloqueados dentro do roteador, e normalmente são utilizados elementos de memória conhecidos por *buffers* (ZEFERINO, 2003).

Os pacotes que trafegam pela rede competem por recursos para que possam atingir o destinatário da comunicação, como canais e buffers, por exemplo. Sendo

assim, então o controle de fluxo é responsável por alocar esses recursos conforme a disponibilidade, permitindo o correto envio e recebimento dos pacotes. De acordo com a política de controle de fluxo, quando um pacote compete por recursos devem existir critérios responsáveis pelas escolhas, como, por exemplo, desviar o caminho de um pacote quando o caminho solicitado está ocupado, a fim de evitar congestionamento e perda de pacotes. Outras opções de decisão são, por exemplo, bloquear a comunicação, armazenando os pacotes em buffers ao longo do caminho (que podem estar na entrada e/ou na saída dos roteadores) ou em último caso, descartar os pacotes (ZEFERINO, 2003).

O controle de fluxo mais comum em redes intra-chip ocorre quando os roteadores possuem buffers no canal de entrada e uma linha de retorno ao transmissor para informar se existe espaço disponível (ZEFERINO, 2003), que pode ser interpretada como um crédito. Outra abordagem semelhante baseia-se no protocolo de *handshake*, em que o emissor informa a intenção de enviar um dado e o receptor confirma a disponibilidade de espaço através de uma linha de validação (*acknowledgement*).

2.1.5. Arbitragem

A arbitragem é responsável por determinar qual canal de entrada pode utilizar um certo canal de saída do roteador de acordo com o tempo. Isso significa que se existe mais de um canal de entrada tentando disputar um mesmo canal de saída, será a arbitragem que escolherá qual canal de entrada terá prioridade para utilizar o canal de saída. A principal diferença entre o roteamento e a arbitragem é que o primeiro define a rota dos pacotes, enquanto que a arbitragem define a prioridade de uso para as rotas que disputam um mesmo caminho.

O mecanismo de arbitragem pode definir diferentes tipos de critérios, que podem ser estáticos ou rotativos (*Round-Robin*). Isso significa que o mecanismo estático pode definir prioridades de modo que, dependendo do tráfego da rede, causem o acontecimento de *starvantion*, pois um determinado canal pode ser sempre favorecido em relação aos outros que solicitam a comunicação dependendo das prioridades estabelecidas. Já o mecanismo de arbitragem rotativo, oferece prioridades diferentes de acordo com a utilização dos caminhos anteriores. Se um determinado canal de entrada A e um certo canal de entrada B usaram recentemente um mesmo canal de saída, então no próximo ciclo de arbitragem a prioridade de uso será dada primeiramente para os canais que ainda não utilizaram aquele canal de saída, ou que utilizaram ele a mais tempo. Por isso a prioridade é dita rotativa, pois num ciclo determinados canais tem alta prioridade na requisição da saída, enquanto no ciclo seguinte eles recebem baixa prioridade de uso daquele canal por terem utilizado ele recentemente.

2.2 Definição de Métricas

Muitas métricas foram adotadas ao longo deste trabalho para comparação dos resultados. Normalmente, elas são métricas conhecidas na literatura, mas a fim de deixar claro para o leitor, a definição de cada parâmetro avaliado neste trabalho é então especificada como consta a seguir:

*Área:*é medida em mm^2 e diz respeito a todo espaço ocupado no chip.

Conectividade: é uma medida em % que diz quantas conexões e fios da rede estão funcionando e podem ser utilizados.

Energia: é definida como a potência dissipada vezes um determinado período de de tempo.

Atraso do Caminho Crítico: é o tempo necessário para enviar uma informação de um ponto ao outro pelo caminho mais complexo do circuito.

Frequência: é o inverso do sinal de relógio, que deve ser no mínimo equivalente ou maior que o atraso do caminho crítico.

Largura de Canal: é o tamanho de bits adotado para cada interconexão. Cada interconexão é composta por *n* fios, e em cada fio apenas 1 bit pode ser transmitido por vez. A largura de canal também pode ser considerada como *largura de banda*.

Latência: é o valor que indica quanto tempo uma informação leva para atingir um alvo, desde que deixa a sua fonte, e é medido em ciclos de execução. A latência é basicamente o valor que se refere a intervalo de tempo que transcorre desde o instante da saída da informação até o primeiro pacote atingir o destino.

Potência: é a capacidade de produção de energia por unidade de tempo. A potência, neste trabalho, é medida para um roteador, para um fio, e para a rede toda é considerado o somatório da potência de todos os roteadores e fios utilizados.

Tempo de Comunicação: é o tempo necessário para enviar toda a informação entre um alvo e um destino. Se 100 pacotes são enviados, então conta-se o tempo necessário para enviar da fonte ao destino todos os 100 pacotes.

2.3 Exemplos de Arquiteturas Encontradas na Literatura e na Indústria

Com base nos conceitos apresentados anteriormente, é possível encontrar muitas arquiteturas de redes intra-chip na literatura, que priorizam diferentes características. A

seguir, algumas delas foram escolhidas e são apresentadas na tabela 2.1, classificada de acordo com os parâmetros apresentados anteriormente na seção 2.1.

Tabela 2.1: Comparação entre redes Intra-Chip.

Rede Intra-Chip	**Topologia**	**Largura do Canal de Comunicação (bits)**	**Roteamento**	**Elementos de Memória**	**Controle de Fluxo**	**Canal Virtual**
ASNOC (JEANG et al., 2006)	Árvore Binária	32	*Wormhole*[§]	Buffers na Entrada	*Handshake*	Não
Æthereal (GOOSSENS E HANSON, 2010)	Grelha/Torus	32	Tabelas	Buffers na Entrada	Baseado em Créditos	Sim
Hermes (MELO, 2005)	Grelha/Torus	16	*Wormhole* XY / Parcialmente Adaptativo	Buffers na Entrada	Baseado em Créditos	Sim
SoCIN (ZEFERINO E SUZIN, 2003)	Grelha/Torus	#	*Wormhole* XY	Buffers na Entrada	*Handshake*	Não
SPIN (GUERRIER and GREINER, 2000)	Árvore Gorda	32	*Wormhole* / Adaptativo	Buffers na Entrada e na Saída	Baseado em Créditos	Não
StNoC (PALERMO et al., 2007)	Anel combinada com Estrela (*Spidergon*)	32	Tabelas	Buffers*	Não cita	Sim
Xpipes (BERTOZZI E BENINI, 2004)	#	#	*Wormhole* / Tabelas	Buffers na Saída	ACK / NACK	#

[§]Não informa o tipo de roteamento. *Não cita onde os buffers estão localizados. #Parametrizável.

De acordo com a tabela 2.1 pode-se observar que cada rede tem suas peculiaridades. ASNOC (JEANG et al., 2006) e SPIN (GUERRIER and GREINER, 2000) por exemplo, possuem como diferencial suas topologias, que são árvore binária e árvore gorda, respectivamente. A rede Æthereal (GOOSSENS E HANSON, 2010) utiliza um misto de chaveamento por circuito e por pacotes, e relaciona alguns compromissos bastante explorados como performance e custos de área. A rede Hermes

(MELO, 2005), desenvolvida pelo GAPH da PUC, aqui mesmo em Porto Alegre, avalia principalmente o impacto do uso de canais virtuais, mostrando que pode reduzir a latência em até 50% para redes de tamanho 8x8, suportando qualidade de serviço.

A rede intra-chip Xpipes (BERTOZZI E BENINI, 2004), uma das mais conhecidas na literatura, trabalha com estágios de *pipeline* e tem por principal objetivo prover alto desempenho e confiabilidade na comunicação através de configurações parametrizáveis em tempo de projeto, além de introduzir técnicas de tolerância a falhas. A StNoC, desenvolvida pela empresa *STMicroelectronics*, também é amplamente conhecida, e tem como grande diferencial a sua topologia *Spidergon* que combina o uso das topologias anel e estrela a fim de reduzir as distâncias na comunicação (PALERMO et al, 2007).

Finalmente, dentre os exemplos apresentados, existe a rede SoCIN (ZEFERINO E SUSIN, 2003) que foi utilizada para empregar as técnicas desenvolvidas neste trabalho e que é apresentada com mais detalhes a seguir.

2.4 Arquitetura Alvo: a Rede SoCIN

Todas as informações disponibilizadas nesta seção são provenientes do trabalho de doutorado do ex-aluno da UFRGS Carlos Alberto Zeferino, apresentado em junho de 2003 (ZEFERINO, 2003). Juntamente com este trabalho, a implementação da rede SoCIN também foi apresentada no Simpósio de Circuitos Integrados e Projeto de Sistemas em 2003, na cidade de São Paulo/Brasil (ZEFERINO E SUSIN, 2003).

A rede SoCIN (*System-on-Chip, Interconnection Network*), é configurada com a topologia grelha ou torus, como mostra a figura 2.4 (a), possibilitando uma melhor tolerância a falhas devido a sua regularidade. Ela também é baseada no controle de fluxo do tipo *handshake*, com roteamento determinístico, chaveamento por pacotes do tipo *wormhole*, arbitragem rotativa e buffers na entrada de cada canal do roteador.

A rede SoCIN é composta pelo roteador RASoC (*Router Architecture for Systems-on-Chip*) (ZEFERINO et al., 2004) que trata-se de um *soft-core* em VHDL parametrizável em três aspectos: largura dos canais de comunicação, profundidade dos buffers e largura da informação de roteamento no cabeçalho do pacote.

As interconexões da rede SoCIN são implementadas por dois canais unidirecionais *simplex*, que possuem *n* bits de dados e dois bits adicionais para o controle do tipo de flit enviado, constituindo assim a largura física do canal (*phit*). O tipo do flit é determinado pelos marcadores de início de pacote (*bop – begin-of-packet*) e fim de pacote (*eop – end-of-packet*), e é classificado em apenas três tipos: cabeçalho (*header*), dados (*payload*) e terminador (*tail*).O primeiro flit do pacote constitui o seu cabeçalho e inclui as informações necessárias ao estabelecimento do caminho do pacote

na rede. Os demais flits incluem a informação a ser transferida pelo pacote, sendo que eles seguem o flit de cabeçalho pela rede. Cada roteador da rede tem capacidade para armazenar poucos flits de um pacote bloqueado, e eles se movem na rede conforme a disponibilidade de alocação dos canais. Quando o flit terminador é transmitido, os canais utilizados naquele caminho são liberados para serem ocupados por flits provenientes de outras fontes. O formato do pacote utilizado é mostrado na figura 2.4 (b).

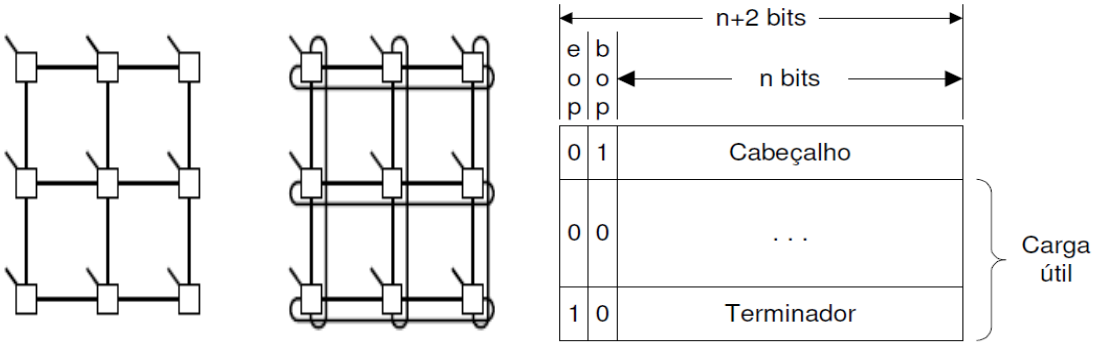


Figura 2.4: (a) Topologia grelha e torus utilizadas pela rede SoCIN e (b) formato do pacote da rede SoCIN (ZEFERINO, 2003).

Além dos *n*+2 bits, cada canal *simplex* inclui um par de sinais necessários ao controle de fluxo (*val* e *ack*), os quais não são contabilizados no cálculo do *phit* da rede, pois não atravessam os roteadores e não são armazenados em seus buffers.

A rede SoCIN utiliza o roteamento XY livre de *deadlock*, em que um pacote deve percorrer totalmente uma linha na direção X até chegar à coluna na qual se situa o destinatário, e em seguida ele deve percorrer essa coluna até atingir o roteador ao qual o destinatário alvo está conectado. Uma vez que um pacote toma a direção Y, ele não pode mais utilizar a direção X.

A rede SoCIN utiliza memorização na entrada, implementada sob a forma de buffers *FIFO* (*First-In*, *First-Out*) e os flits são lidos das partições na mesma ordem em que são escritos, através do uso de um buffer circular.

Nos roteadores da rede SoCIN, cada entrada possui um buffer *FIFO* com capacidade de armazenar *p* flits de *n*+2 bits, sendo *p* um parâmetro ajustado em tempo de projeto em função dos requisitos do sistema. Quanto maior é o seu valor, mais profundos são os buffers e menor é a contenção na rede. Para este trabalho, o valor de *p* foi definido em 4 e o valor de *n* ficou definido em 8 e posteriormente em 32. A técnica de controle de fluxo utilizada na rede SoCIN é baseada no protocolo de *handshake*, no qual o emissor informa a intenção de enviar um dado ao receptor através de uma linha de validação e o receptor confirma a disponibilidade de espaço em buffer para receber esse dado através de uma linha de reconhecimento (*acknowledgement*). De fato esse sinal de reconhecimento sinaliza ao emissor que o flit por ele injetado no canal de comunicação será consumido pelo receptor no ciclo seguinte do relógio. O reconhecimento é dado sob a condição de haver uma validação no canal e se o buffer de

entrada do receptor não estiver cheio. Esse mecanismo é bastante conservador e garante que nenhum flit seja descartado, pois a transmissão só é realizada após um acordo entre os envolvidos na negociação (emissor e receptor).

A figura 2.5 mostra uma abstração da arquitetura do roteador RASoC com 5 portas de entrada e saída, onde é possível ver uma generalização do mecanismo de *handshake*, os 4 slots de buffer, o árbitro (simbolizado por A) e o *crossbar* onde todos os canais se conectam (exceto a si mesmo).

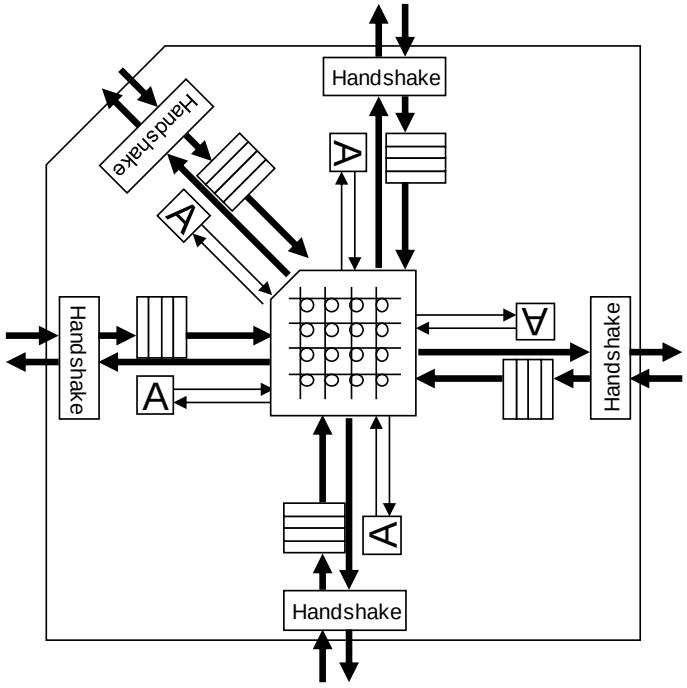


Figura 2.5: Arquitetura básica do roteador RASoC (ZEFERINO et al., 2004).

A rede SoCIN utiliza um esquema de arbitragem distribuída no qual cada porta de um roteador possui um árbitro que seleciona o buffer de entrada do roteador a ser conectado ao canal de saída dessa porta. Os árbitros utilizam um critério baseado em prioridades dinâmicas rotativas (*Round-Robin*) garantindo que nenhum pacote sofrerá com o problema de *starvation.* O árbitro de cada canal de saída é integrado junto a um circuito controlador de saída. Esse circuito é responsável por controlar a temporização dos sinais de confirmação que comandam os circuitos de chaveamento. A conexão entre o buffer de entrada selecionado e o canal de saída deve ser mantida até que o flit terminador seja transferido pelo canal.

3. TRABALHOS RELACIONADOS: TOLERÂNCIA A FALHAS EM REDES INTRA-CHIP

A integração de muitos componentes em um único chip trouxe severas preocupações para os projetistas de circuitos integrados com relação a confiabilidade e ao rendimento dos chips produzidos. Técnicas de tolerância a falhas estão sendo estudadas amplamente por diversos grupos de pesquisa para proporcionar a capacidade de um circuito tolerar diferentes tipos de falhas, em diferentes áreas de atuação (aeronáutica, medicina, satélites, etc.). Muitos trabalhos sobre este tema, com o foco voltado em geral para circuitos com redes intra-chip e com a topologia grelha ou torus, foram desenvolvidos aqui mesmo em Porto Alegre por grupos de pesquisa próprios da Universidade Federal do Rio Grande do Sul, como (FRANTZ et al., 2006), (CONCATTO et al., 2009), (HERVÉ et al., 2009), (KOLOGESKI et al., 2010) e (BRAGA et al., 2010), e por grupos próximos, como é o caso da rede Hermes desenvolvida pelo Grupo de Apoio ao Projeto e Hardware da Pontifícia Universidade Católica do Rio Grande do Sul (MELO et al, 2005).

Com a miniaturização dos transistores, a manufatura dos circuitos tornou-se muito delicada, sujeita a defeitos permanentes de fabricação oriundos de diversas fontes. Além disso, existem fontes de falhas transientes como interferências, pulsos eletromagnéticos, radiação e partículas provenientes do espaço e do sol que podem atingir o circuito e interferir na resposta final alterando o resultado esperado. Estes problemas tornaram-se mais comuns com o desenvolvimento da tecnologia, pois normalmente em transistores mais antigos, com largura de canal maior, eles não eram suficientemente fortes para causar danos efetivos alterando a condução.

Como já citado anteriormente, de acordo com as projeções de (FURBER, 2006), chips com 100 bilhões de transistores terão em torno de 20% dos transistores produzidos com defeitos já no processo de manufatura, e outros 10% irão falhar no primeiro ano de operação.Os autores (DEHON e NAEIMI, 2005) também afirmam que a taxa de fios e conexões defeituosas tende a ser de aproximadamente 1% até 15%. Sendo assim, não restam dúvidas de que técnicas para proteger os circuitos desenvolvidos serão fundamentais para garantir a correta funcionalidade dos circuitos projetados atualmente.

3.1 Definição das Falhas

Defeitos podem acontecer em qualquer região de um circuito, como, por exemplo entre as interconexões de uma rede e nos transistores da lógica projetada, muitas vezes provenientes do depósito de partículas ou impurezas indesejadas no processo de fabricação. Os modelos de falhas mais comumente utilizados para falhas de efeito permanente são o de curto circuito, circuito aberto e de *stuck-at* 0 e *stuck-at* 1 (VENKATARAMAN and DRUMMONDS, 2000). Este trabalho foca em falhas de interconexão, mas algumas das técnicas de tolerância apresentadas poderão também ser utilizadas para tolerar falhas em roteadores e em núcleos, de acordo com a maneira como são empregadas.

A fim de exemplificar as falhas consideradas neste trabalho, nas interconexões da rede, a aplicação MPEG4 descrita por (BERTOZZI et al., 2005) foi mapeada em uma rede 4x3 com topologia torus, e o conjunto de falhas que está sujeito a acontecer na rede intra-chip é descrito e exemplificado como consta na figura 3.1. Falhas do tipo *intralink* e *interlink* são consideradas.

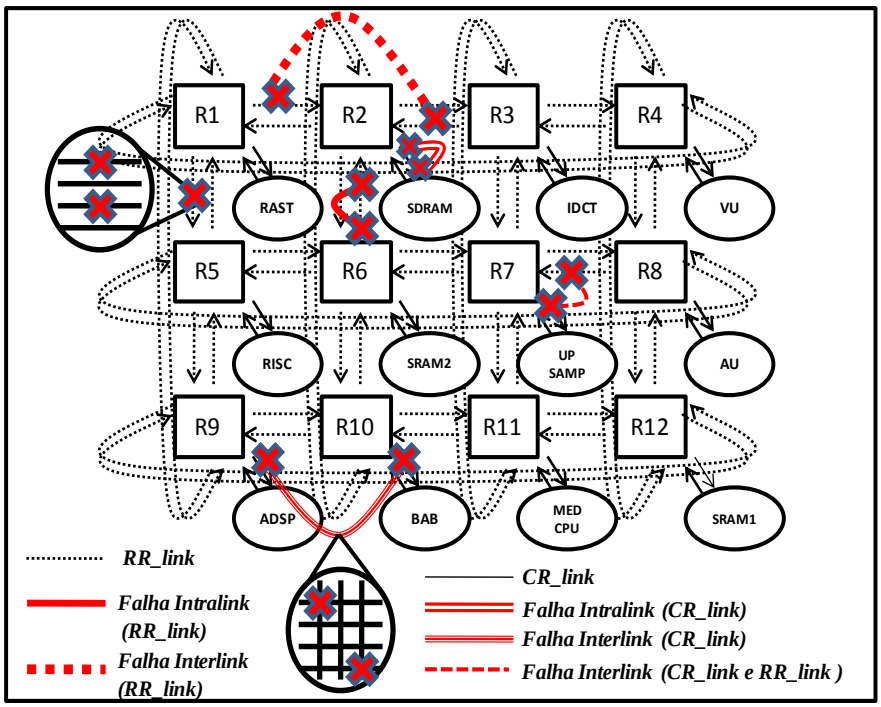


Figura 3.1: MPEG4 mapeado em uma rede 4x3 torus com o exemplo de um conjunto de falhas considerado entre as interconexões.

Falhas *intralink* acontecem quando o fio agressor e fio vítima estão dentro de uma mesma interconexão, e falhas *interlink* acontecem quando o fio agressor e o fio vítima estão em interconexões diferentes. Então, falhas podem ser consideradas *intralink* quando elas ocorrem apenas dentro de uma interconexão do tipo *RR_link*

(interconexões que ligam um roteador a outro roteador) ou de uma interconexão do tipo *CR_link* (interconexões que ligam roteadores e núcleos). Para uma falha ser considerada *interlink* é necessário que ela aconteça entre duas ou mais interconexões diferentes, podendo ser entre dois ou mais *CR_links*, entre dois ou mais *RR_links* e também entre dois ou mais *RR_links* combinados com *CR_links*.

Múltiplos defeitos podem ser quaisquer combinações dos casos de falhas apresentados, desde que exista mais de uma falha no conjunto total de interconexões.

3.2 Trabalhos Relacionados

Os trabalhos relacionados podem ser classificados principalmente pela forma como as técnicas de tolerância a falhas atuam para prover confiabilidade. As técnicas podem ser divididas em dois grandes grupos, de acordo com o tipo de diagnóstico de falhas que é utilizado. Existem técnicas que são classificadas como dinâmicas, que são aquelas que estão sempre atuando no hardware para protegê-lo, principalmente contra falhas transientes. Neste caso, o diagnóstico e detecção são feitos em tempo de execução, e a correção do dado pode ou não ser feita simultaneamente. Alguns exemplos de técnicas dinâmicas utilizadas são o código de Hamming, que na sua versão mais simples permite detectar até duas falhas e corrigir apenas uma em cada interconexão. A paridade também é um exemplo, que permite detectar a alteração dos dados, normalmente sem detectar a posição onde a falha ocorreu, bem como a redundância de hardware como é o caso da técnica conhecida por TMR, que significa tripla redundância modular, na qual um votador é utilizado para escolher os dados corretos e rejeitar os dados alterados devido a influência de uma falha. Porém, o preço que se paga é um alto custo de área extra, e consequentemente um aumento excessivo de potência, porque as técnicas utilizadas são desenvolvidas em tempo de projeto e estão sempre ativas na comunicação para encontrar falhas em tempo de execução, pois elas precisam ser executadas constantemente, mostrando um alto impacto na eficiência energética de qualquer circuito que as utilize.

O outro tipo de técnica que é empregada em circuitos são técnicas estáticas de detecção e diagnóstico. Este tipo de técnica necessita previamente da informação de localização das falhas, para poder configurar suas estruturas de tolerância a falhas. Quando nenhuma falha é diagnosticada na rede através de um teste prévio (ou teste de manufatura) então o hardware adicional pode ser ignorado a fim de evitar excessos em potência. Normalmente, existem fios adicionais ou componentes extras que são utilizados, conhecidos como fios ou componentes *spare*, ou algum roteamento adaptativo é utilizado para evitar os locais defeituosos da rede. A área necessária para técnicas estáticas pode também ser considerada grande, porém os ganhos ocorrem quando o hardware de tolerância a falhas não precisa ser completamente utilizado, evitando gastos desnecessários em potência, por exemplo.

Alguns trabalhos que lidam com a detecção estática e o diagnóstico de falhas nas interconexões são: (CONCATTO et al., 2009), (HERVÉ et al., 2009) e o trabalho de (YANG e PAPACHRISTOU, 2009). Os dois primeiros trabalhos lidam com a detecção e o diagnóstico de falhas nas interconexões através da propagação de vetores de teste pela rede, enquanto que o segundo apresenta um método para detectar defeitos em interconexões de SoCs, usando teste I_{DDT} (que analisa a variação da corrente dinâmica), testes de atraso e *boudary scan*.

3.2.1 Técnicas Dinâmicas

Quando o código de Hamming é utilizado, existe um impacto direto no desempenho do circuito, pois os codificadores são compostos por uma cascata de portas "OU exclusivo" inseridas no caminho crítico da comunicação, que estão sempre ativas para garantir a tolerância a falhas. No trabalho de (FRANTZ et al., 2006) a combinação de algumas técnicas usando código de Hamming foi proposta para proteger os buffers, o roteador e também as interconexões que ligam um roteador ao outro. Como mostra a figura 3.2, uma das implementações de FRANTZ et al. (2006) consiste em dados que são codificados ao entrar no buffer, necessitando de um buffer maior para o armazenamento, decodificados ao deixar o buffer, e passam pela lógica interna do roteador, composta pelo roteamento e pela arbitragem, sem qualquer proteção. Novamente, ao deixarem o roteador, os dados são codificados para atravessar a interconexão e decodificados ao atingirem um novo canal de entrada, necessitando então de fios extras para a transmissão. Os autores protegem o dado de apenas uma única falha em cada flit, focando em falhas transientes no buffer e crosstalk nas interconexões. Como mostra a figura 3.3, casos em que existem múltiplas falhas não podem ser tratados pelo trabalho de FRANTZ et al. (2006), sem falar na necessidade da utilização de fios extras para transmitir a codificação. Resultados reportados mostram uma penalidade na frequência de 32% para uma tecnologia de 180 nm, e mais de 50% em área extra sem incluir o aumento de fios nas interconexões.

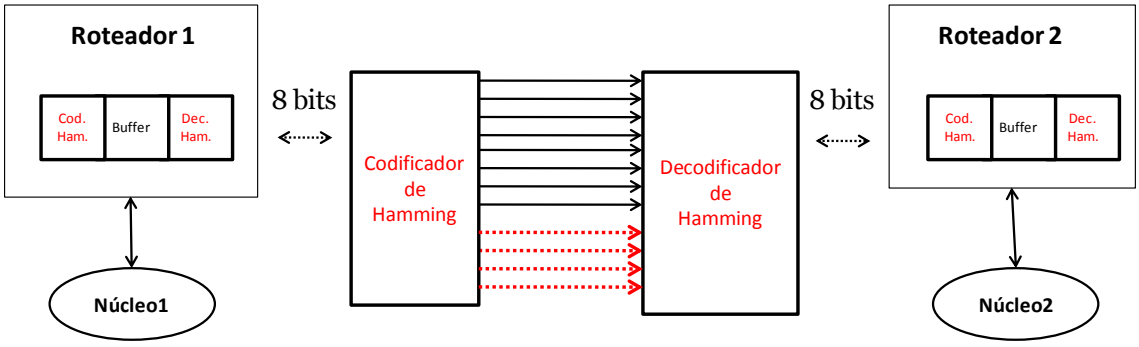


Figura 3.2: Esquema de proteção contra falhas utilizando código de Hamming (FRANTZ et al., 2006).

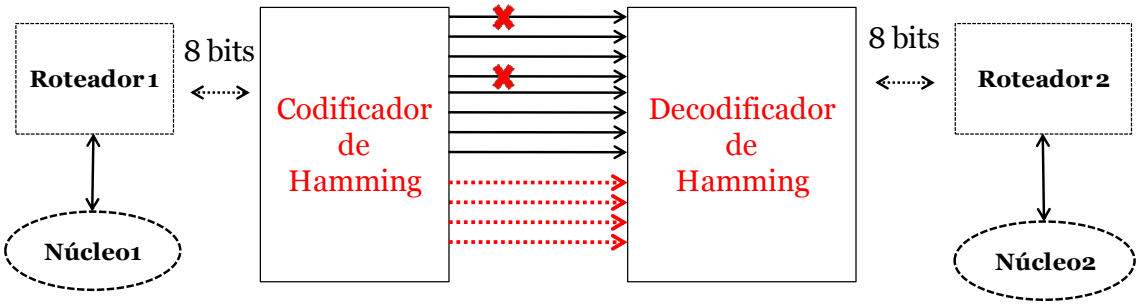


Figura 3.3: Situação de falha nas interconexões que o código de Hamming não pode lidar.

Em (LEHTONEN et al., 2007) um trabalho que apresenta a combinação de diferentes métodos para alcançar tolerância a todos tipos de falhas nas interconexões foi apresentado para a tecnologia de 130 nm. A técnica consiste em usar o código de Hamming em cada metade dos dados, e a retransmissão pode ser utilizada quando Hamming não é suficiente para a correção dos dados com falha. Mesmo assim, se existem múltiplas falhas em alguma parte da interconexão, a retransmissão pode não ser suficiente para manter a confiabilidade dos dados, como a figura 3.4 mostra. Para proteger os links que fazem *handshake*, TMR é utilizado. As principais desvantagens deste método são a quantidade extra de área e o consumo excessivo de potência conforme apresentado nos resultados dos autores. No final, a área do roteador torna-se mais de 3 vezes maior do que o roteador não protegido, e a latência também é incrementada em quase 4 vezes. Além disso, existe a degradação do desempenho na rede devido ao acréscimo de hardware no caminho crítico, e o impacto na energia pode ser em mais de 100%.

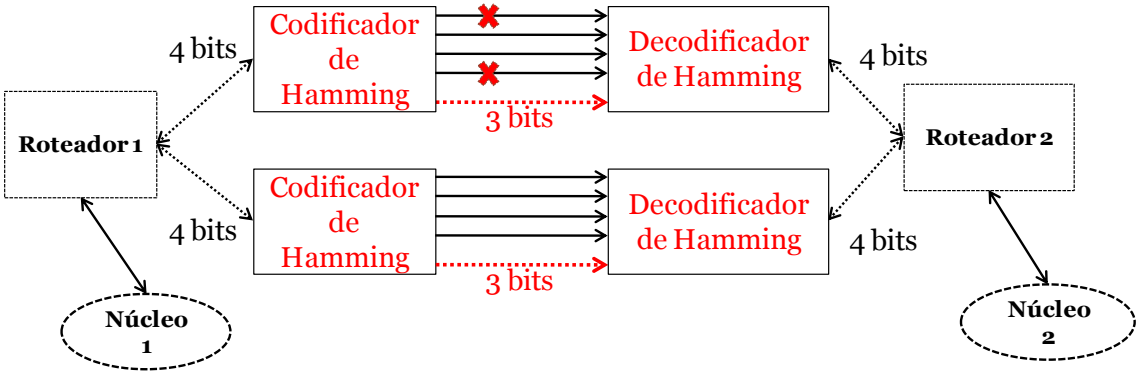


Figura 3.4: Situação de falha nas interconexões em que nem mesmo a retransmissão dos dados de LEHTONEN et al. (2007) não pode lidar.

O trabalho de (BRAGA et al., 2010) mostra uma proposta semelhante a de (LEHTONEN et al., 2007) para proteger as interconexões contra crosstalk e falhas permanentes, utilizando a tecnologia de 180 nm. A proposta consiste em transmitir um bit de paridade para cada metade do dado, a fim de permitir a detecção de uma falha no link. Na presença de uma falha, a metade dos dados que se encontra defeituosa é

retransmitida ocupando as duas metades do link, enviando novamente um bit de paridade para cada metade. Na retransmissão, a metade que for recebida sem erros é escolhida. O impacto no desempenho é mínimo, porque a penalidade na latência somente é dobrada quando o dado é retransmitido, ou seja, somente quando existem falhas. Este método não necessita de um diagnóstico prévio das falhas, e utiliza fios extras para transmitir os bits de paridade, junto com TMR e duplicação para garantir a integridade dos sinais de controle da comunicação (*val* e *ack*). Em relação à implementação do roteador sem tolerância a falhas, BRAGA apresentou um impacto de 36% na frequência máxima, e quase 22% de área extra utilizada, considerando que pode detectar e corrigir (através da retransmissão) múltiplas falhas nas interconexões, enquanto que o uso do código de Hamming protege cada interconexão de apenas uma única falha, com o mesmo impacto na frequência e 6% de área extra como os autores mostram. Porém, BRAGA tolera crosstalk e apenas uma única falha permanente em cada interconexão. Se, por exemplo, existem 2 falhas permanentes em uma interconexão, sendo cada uma delas localizada em uma das metades do fio como mostra a figura 3.5, significa que o bit de paridade sempre estará errado, e mesmo utilizando a retransmissão não será possível ter um dado sem a presença de erro.

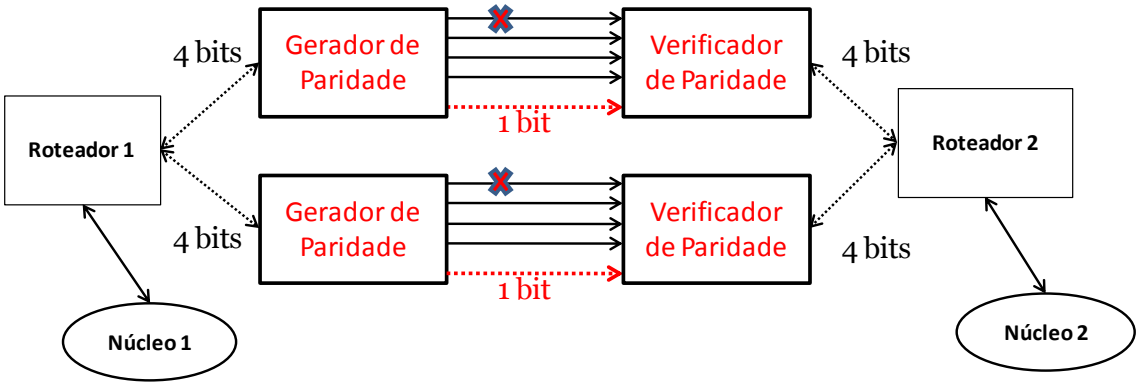


Figura 3.5:Situação de falha nas interconexões em que (BRAGA et al., 2010) não pode lidar com o uso de retransmissão e paridade.

Em (GANGULY et al., 2009) os autores apresentam uma proposta com código de Hamming onde as interconexões são completamente duplicadas, como mostra a figura 3.6, permitindo que até três falhas sejam toleradas, e quatro falhas sejam detectadas. Isso porque quando duas falhas acontecem em um dos conjuntos de dados ele torna-se inutilizável, enquanto que o outro conjunto de dados (devido à duplicação) pode ter até uma falha e corrigi-la a fim de obter o dado original livre de falhas. Quando acontecem duas falhas em cada conjunto, então a implementação permite apenas detectá-las, já que o uso de Hamming permite correção apenas de falha única para cada conjunto de fios. Com base nas previsões de (DEHON e NAEIMI, 2005), tolerar apenas três falhas nas interconexões pode não ser suficiente para prover uma comunicação adequada na rede com canal de 32 bits, já que até 15% dos fios poderão estar defeituosos.

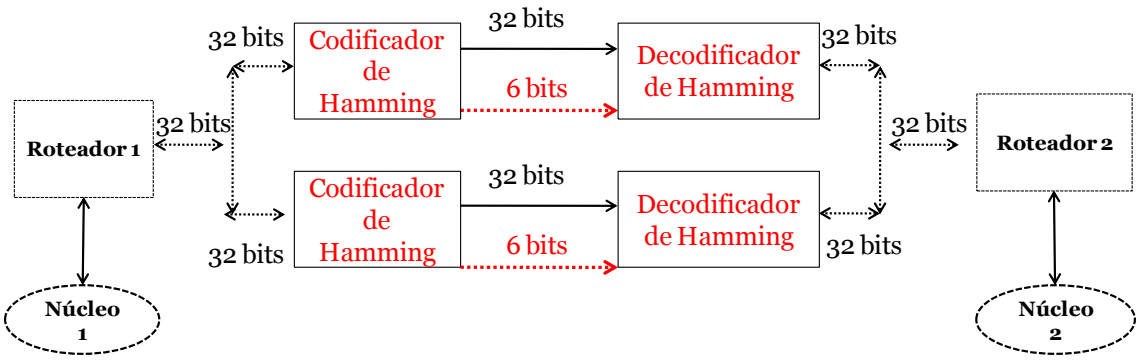


Figura 3.6: Proposta de (GANGULY et al., 2009) com uso de código de Hamming e interconexões duplicadas.

O grande diferencial do trabalho apresentado por (GANGULY et al., 2009) é que os autores reduzem a tensão nos fios para a transmissão, reduzindo também a probabilidade de erros devido ao crosstalk. Desta forma, eles podem reduzir até 46,6% de potência em relação ao trabalho original sem qualquer proteção, com 22% de área extra em relação ao roteador, sem considerar as interconexões extras. Porém, o hardware necessário para o uso de código Hamming foi completamente duplicado, assim como todas as interconexões, mostrando que a solução não leva em conta a penalidade de ter o dobro de interconexões na rede, e que não pode tolerar mais do que 3 falhas em cada interconexão (a correção de 3 falhas somente é possível devido a duplicação de cada interconexão).

3.2.2 Técnicas Estáticas

Todos os trabalhos apresentados até então possuem um restrição em comum: um limitado número de falhas para serem toleradas por cada interconexão. Para solucionar esse problema, (PALESI et al., 2010) propõem o uso de links parcialmente defeituosos (com falhas em apenas alguns fios). Os autores justificam o uso das interconexões parcialmente defeituosas principalmente quando existe a presença de um alto tráfego de dados na rede, proporcionando uma melhor distribuição do tráfego. Os autores mostram que usar a interconexão parcialmente defeituosa é frequentemente melhor do que evitá-la através de uma função de roteamento. A capacidade dos links para tolerar falhas permanentes pode ser dividida em 25%, 50%, 75% e 100%, de acordo com a localização das falhas. Cinco estratégias básicas são utilizadas, sendo que elas podem ser resumidas basicamente na combinação de três estratégias: a primeira evita completamente links defeituosos através do uso de uma função de roteamento específica, a segunda pode ou não utilizar links parcialmente defeituosos, e a última estratégia sempre irá utilizar links parcialmente defeituosos. A figura 3.7 mostra uma idéia geral de como é estruturado o envio e o recebimento dos dados ocupando apenas partes não defeituosas da interconexão (fios livres de falhas), para um link de 64 bits.

Porém, a solução de (PALESI et al., 2010) considera que as falhas estejam todas concentradas dentro do mesmo grupo de fios (por exemplo, um link de 32 bits dividido em 4 grupos de 8 bits), o que não corresponde a um fato completamente real, já que as falhas podem estar distribuídas entre todo o link de comunicação, como mostra a figura 3.8 para uma interconexão de 64 bits.

A área extra necessária para as implementações de (PALESI et al., 2010) varia de 15% até 21%. A frequência de operação não foi afetada pelo acréscimo da lógica, uma vez que ela é limitada pela lógica de controle e pelo acesso a tabela de roteamento. O consumo de potência extra oscila em torno de 5% a 8%.

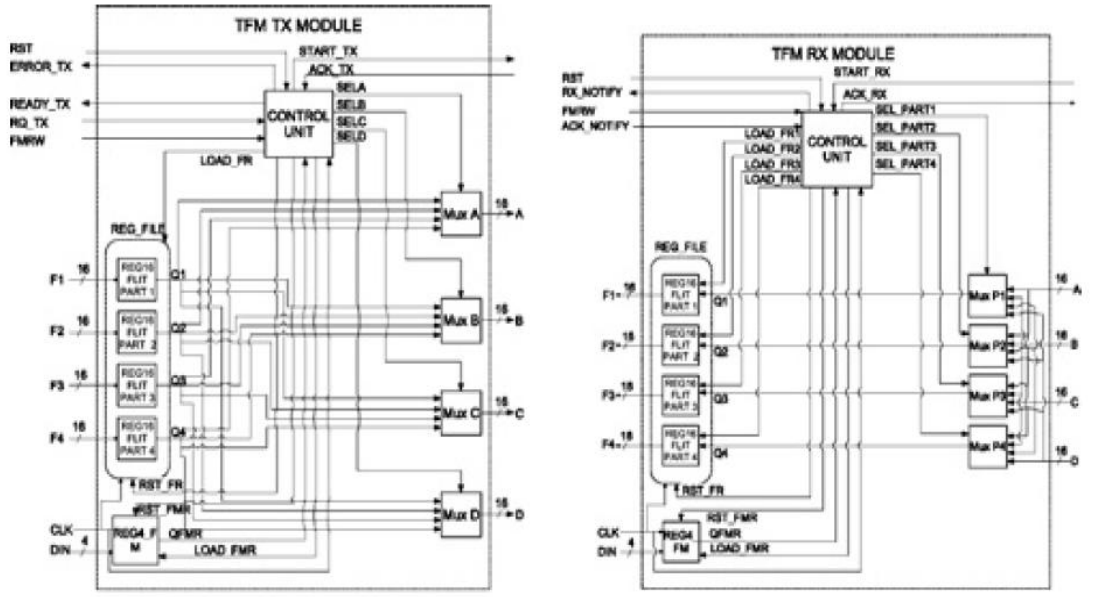


Figura 3.7: Diagrama de blocos do emissor e receptor desenvolvido por (PALESI et al., 2010).

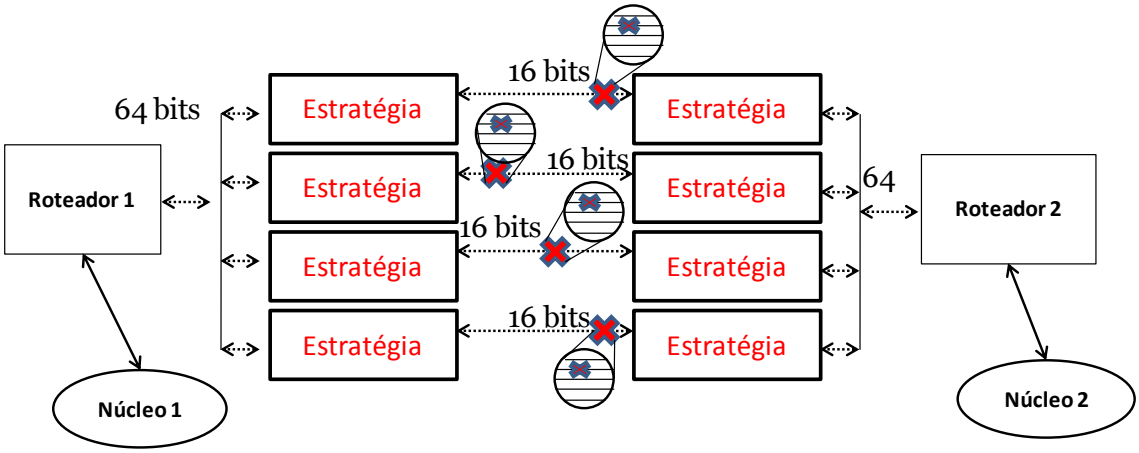


Figura 3.8: Caso que a proposta de (PALESI et al., 2010) necessita evitar a interconexão defeituosa e utilizar uma função roteamento específica para atingir o destino.

Outros trabalhos na literatura utilizam estratégias baseadas nas funções de roteamento para lidar com interconexões defeituosas e evitar roteadores defeituosos,

como é o caso de (SCHONWALD et al., 2007) e (KOIBUCHI et al., 2008). As estratégias por eles utilizadas implicam em uma latência relativamente baixa. Contudo, o uso de tabelas e canal virtual é necessário para evitar-se o problema de *deadlock* na rede, sendo que ambas as opções são sinônimos de área e potência extra. Os autores em (CONCATTO et al., 2009) propõem o uso de uma estratégia com um roteamento parcialmente adaptativo para lidar com interconexões defeituosas tendo-se uma mínima mudança no caminho do roteamento XY, através do uso da topologia torus. Consequentemente, canais virtuais e tabelas não são utilizados, e a área extra necessária é consideravelmente pequena (menos de 1%). Contudo, por ser um algoritmo parcialmente adaptativo, existem alguns casos em que um caminho livre de falhas não pode ser encontrado, especialmente na presença de múltiplas falhas. De acordo com os resultados de (CONCATTO, 2009), existem 34% de casos que não podem ser solucionados apenas com o uso do algoritmo parcialmente adaptativo, para uma rede 3x4, e este percentual pode ser muito maior na presença de múltiplas falhas. A figura 3.9 (a) mostra uma situação que não pode ser solucionada: quando as falhas se encontram em alguma interconexão da rede entre o roteador e o núcleo, desde que esta interconexão seja o único meio de comunicação entre eles. Na figura 3.9 (b) algumas das situações em que o roteador deixa de ser utilizado são apresentadas: quando ambas as entradas ou saídas de uma mesma direção encontram-se defeituosas não permitindo o acesso ao roteador através daquela direção.

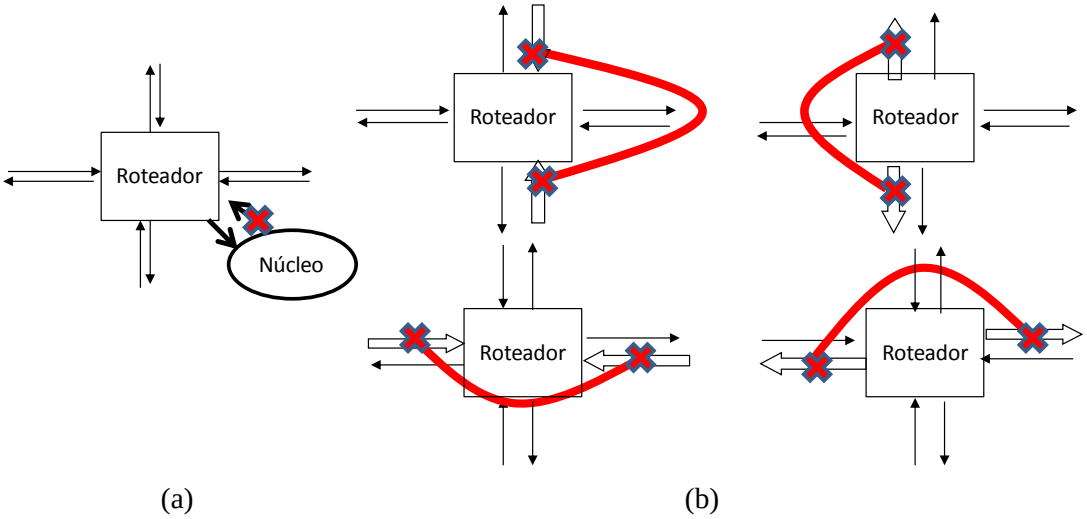


Figura 3.9: Casos limitados de falhas que o trabalho de CONCATTO et al. (2009) não pode lidar: (a) inutilizando um roteador ou (b) não acessando um núcleo.

Os trabalhos de (TORNERO et al., 2009) e (CHOUDHURY et al., 2009) combinam roteamento adaptativo e mapeamento para incrementar a confiabilidade das redes intra-chip. Ambos os trabalhos apresentam uma estratégia que leva em conta o grafo da aplicação, a probabilidade de falhas e o roteamento. O principal objetivo é obter o conjunto Pareto com funções de roteamento customizadas que minimizam a latência e maximizam a confiabilidade da aplicação. Ambas as propostas não podem

lidar com falhas entre um roteador e um núcleo, optando por deixar um roteador livre de núcleo ou inutilizar o núcleo conectado ao roteador em caso de falha no link que os liga. Na presença de falhas nas interconexões, (CHOUDHURY et al., 2009) pode lidar com 96% dos casos em uma rede intra-chip 3x4 com topologia grelha, por exemplo. A eficiência de ambos os trabalhos pode ser ainda reduzida para 65% se no total de interconexões com possibilidade de falhas forem incluídos aqueles links que ligam núcleos e roteadores, para uma rede 3x4 com topologia grelha.

No trabalho de (KAKOEE et al., 2011) a redundância é utilizada em alguns componentes no roteador e nas interconexões para prover confiabilidade. Uma estrutura BIST (*Built-In Self-Test*) foi incluída na implementação para prover o diagnóstico da falhas. Os resultados de área mostram um aumento de 12,5 até 15,5%, de acordo com o número de buffers utilizados, para uma interconexão que corresponde em torno de 5% da área total da rede. Porém, este trabalho apresenta um baixo percentual de conectividade (links funcionando na rede) quando o número de falhas cresce consideravelmente na rede, chegando a apenas 30% de conectividade quando existem até 100 falhas numa rede intra-chip 8x8.

De acordo com as previsões de falhas de (DEHON e NAEIMI, 2005) e também de (FURBER, 2006), é necessário levar em conta o fato de que poderá existir pelo menos uma falha em cada interconexão, já que a proporção de falhas nas interconexões pode chegar a 15%. Sendo assim, algoritmos de roteamento que podem ser adaptados de acordo com as falhas e uma simples redundância não serão suficientes para prover uma comunicação confiável, considerando que todas as interconexões da rede podem estar com algum tipo de falha tornando a conectividade nula, se cada defeito inutilizar toda uma conexão. Desta forma, está claro que ainda existe a necessidade de resolver o problema para múltiplas falhas nas interconexões de uma rede intra-chip. Soluções que detectam e diagnosticam falhas dinamicamente não cobrem uma quantidade significativa de falhas, e podem também necessitar de uma grande quantidade de área extra, resultando num elevado consumo de potência e incrementando o tempo de comunicação, uma vez que essas estruturas são adicionadas no caminho crítico e estão sempre ativas. Em (SCHONWALD et al., 2007) e (KOIBUCHI et al., 2008) os autores não informam detalhes de como é feita a detecção das falhas, enquanto que (PALESI et al., 2010) e (CONCATTO et al., 2009) afirmam a necessidade de um diagnóstico prévio para descobrir a exata localização das falhas. Estruturas que utilizam mecanismos de diagnóstico prévio podem garantir melhores comprometimentos para as arquiteturas em que são inseridas, pois estruturas de tolerância a falhas (TF) podem ser configuradas para exercer ou não a proteção, conforme a necessidade do circuito. Quando as falhas não existem, essas estruturas podem ser desativadas a fim de economizar potência e reduzir a energia, enquanto que soluções como código de Hamming estão sempre ativas e não podem proteger múltiplas falhas em uma mesma interconexão.

Na tabela 3.1 encontram-se todos os trabalhos relacionados listados de acordo com o ano da publicação, para obter-se um comparativo mais específico e resumido das técnicas apresentadas neste capítulo. Com base nos trabalhos apresentados, está claro

que ainda existe a necessidade de prover soluções tolerantes a falhas com baixo consumo de potência e um desempenho considerável. Outro fator relevante é o problema de múltiplas falhas concentradas em um mesmo link, algo muito provável de acontecer de acordo com previsões para o avanço da tecnologia. Sendo assim, nos próximos capítulos serão apresentadas as idéias deste trabalho, onde se pretende desenvolver um mecanismo adaptativo utilizado apenas na presença de falhas, com a intenção de reduzir a potência e a energia em relação aos trabalhos já propostos na literatura. A idéia do trabalho desta dissertação consta na última linha da tabela 3.1.

Tabela 3.1: Comparação entre os trabalhos relacionados.

Autores e Ano de Publicação	**Necessita Teste e Diagnóstico**	**Técnica Utilizada**	**Limitações/ Características**
FRANTZ et al. (2006)	Não	Código de Hamming	Tolera apenas uma falha por link
SCHONWALD et al. (2007)	Sim	Algoritmo de roteamento e canal virtual	Falhas entre roteador e núcleo não podem ser solucionadas; potência excessiva
LEHTONEN et al. (2007)	Não	Código de Hamming, divisão dos dados e retransmissão	Área e Latência são fortemente prejudicadas, corrige até 1 falha em cada metade do link
KOIBUCHI et al. (2008)	Sim	Algoritmo de roteamento e tabelas	Falhas entre roteador e núcleo não podem ser solucionadas; potência excessiva
GANGULY et al. (2009)	Não	Código de Hamming, duplicação dos Links, redução da voltagem	Pode corrigir até 3 falhas por link
CONCATTO et al. (2009)	Sim	Algoritmo de roteamento	Falhas em ambos os canais de um roteador, na mesma direção, não podem ser solucionadas, nem entre roteador e núcleo
TORNERO et al. (2009)	Sim	Algoritmo de roteamento, probabilidade de falhas, mapeamento	Não pode lidar com falhas entre roteador e núcleo
CHOUDHURY et al. (2009)	Sim	Algoritmo de roteamento, probabilidade de falhas, mapeamento	Não pode lidar com falhas entre roteador e núcleo
BRAGA et al. (2010)	Não	Paridade, divisão dos dados e retransmissão	Área extra, tolera apenas 1 falha em cada metade do link
PALESI et al. (2010)	Sim	Divisão de dados para evitar grupos de fios falhados	Considera falhas em um mesmo conjunto de fios, o que não reflete a realidade
KAKOEE et al. (2011)	Sim	Duplicação dos componentes	Mantém a conectividade para poucas falhas
KOLOGESKI et al. (2011)	Sim	Divisão de dados, algoritmo de roteamento e remapeamento	Alta conectividade; tolera links com até 50% de fios defeituosos

4. ABORDAGENS E ESTRATÉGIAS PROPOSTAS

Uma nova combinação de estratégias é apresentada nesta seção para lidar com múltiplas falhas nas interconexões de uma rede intra-chip. Com a abordagem desenvolvida é possível tolerar até 50% de fios defeituosos em cada interconexão, sem haver um considerável prejuízo para a rede em termos de área extra, desempenho, potência e energia, como será mostrado nas seções posteriores, para canais de 8 e 32 bits. O método desenvolvido pode garantir a funcionalidade da rede com múltiplos defeitos em qualquer interconexão e também com múltiplas interconexões defeituosas. A técnica proposta utiliza as informações de um teste previamente realizado ou do teste de manufatura para configurar os recursos que permitem tolerar as falhas na rede. Sendo assim, quando nenhuma falha é detectada, os recursos podem ser desligados a fim de minimizar a energia. Isso é possível porque a técnica de *sleep transistor* permite cortar a alimentação de partes do circuito com a inserção de um transistor entre o circuito e a fonte de alimentação (LONG e HE, 2004) (SHI e HOWARD, 2006).

A técnica proposta neste trabalho consiste na combinação do roteamento adaptativo proposto por (CONCATTO et al., 2009) e na divisão dos dados ocupando 50% dos fios livres de falhas das interconexões que foram consideradas defeituosas, deixando sem utilização aqueles fios considerados defeituosos e aproveitando apenas os fios que podem proporcionar uma comunicação de forma correta. Essa abordagem evita que fios adicionais nas interconexões sejam necessários e minimiza a quantidade de hardware adicional no caminho crítico da comunicação. Ambas as estratégias são também combinadas com a técnica de remapeamento, que pode ser facilmente aplicada em redes com núcleos homogêneos. De uma maneira muito simples, o peso das comunicações pode ser calculado de acordo com a intensidade do tráfego em cada canal, e núcleos com baixa comunicação podem ser facilmente mapeados para regiões com falhas, a fim de minimizar os prejuízos na comunicação. Quando os núcleos são heterogêneos, é preciso analisar a necessidade de componentes redundantes na rede, pois a replicação de algumas regiões da rede será necessária para permitir o remapeamento. Se o remapeamento não for desejado devido aos custos que traz para o projeto, apenas as técnicas de roteamento adaptativo e divisão de dados podem ser utilizadas.

A figura 4.1 ilustra um fluxograma que compreende as etapas deste trabalho. Inicialmente, considera-se que o teste das interconexões é realizado com base nos trabalhos de (CONCATTO et al., 2009), (HERVÉ et al., 2009) e de (YANG e PAPACHRISTOU, 2009). Após a detecção e o diagnóstico das falhas, os vetores de teste utilizados por cada roteador indicam se existem ou não falhas na rede e permitem então a configuração dos registradores, utilizados para controle e sinalização das falhas, a fim de que a rede tenha conhecimento dos canais defeituosos.

A primeira técnica aplicada em caso de falha em algum *RR_link* será a técnica de roteamento adaptativo, devido à sua simplicidade e ao seu reduzido impacto, tanto no novo caminho adotado para a comunicação quanto em termos de impacto na área, potência e desempenho, como será mostrado posteriormente. Necessariamente, a técnica de roteamento adaptativo utiliza a topologia torus. Para os casos em que o roteamento adaptativo não pode ser utilizado (outras topologias ou falhas que não são solucionadas pela estratégia, como apresentado no capítulo anterior), a solução de dividir os dados para enviá-los ocupando apenas metade da interconexão é empregada, como será mostrado na seção 6.

A divisão de dados neste trabalho é a única maneira de lidar com a comunicação que precisa utilizar necessariamente o caminho defeituoso, e por isso ela tem a finalidade de manter a conectividade, ou seja, ela evita o isolamento dos núcleos através da região defeituosa (no caso de falhas no *RC_link*). Para casos de falha em algum *RC_link*, a divisão de dados sempre será aplicada, e a estratégia simplificada de remapear as tarefas somente é possível quando núcleos homogêneos podem trocar de posição sem prejuízo algum para a rede (prejuízos em termos de reorganização das tarefas, impacto em área, latência, etc). Para núcleos heterogêneos, com dimensões variadas, a estratégia proposta somente pode ser possível se alguma redundância for utilizada na rede, ou se os núcleos forem mapeados dentro de processadores, por exemplo. Assim, como será mostrado posteriormente na seção 7, existem variações de mapeamento de acordo com a confiabilidade desejada que podem ser pré-fabricadas e utilizadas de acordo com a distribuição das falhas na rede.

Então, o remapeamento permite encontrar um núcleo com mínimo impacto para realizar a troca de posição com o núcleo situado no local defeituoso, mas não evita que a divisão de dados seja utilizada no caso de um *RC_link* com falha. Com a combinação da divisão de dados e do remapeamento, a intenção é utilizar as interconexões mesmo que defeituosas, embora com uma limitação devido ao reduzido número de fios livres de falhas empregados na comunicação, a fim de minimizar os custos apresentados nos trabalhos relacionados.

Por fim, pode-se dizer que existem casos em que todas as estratégias podem ser combinadas para evitar situações de falhas nas interconexões. A idéia deste trabalho resume-se então em combinar as estratégias apresentadas conforme o fluxo da figura 4.1 a fim de tolerar múltiplos fios defeituosos em cada interconexão e múltiplas interconexões defeituosas, com o objetivo de reduzir os custos extras na implementação.

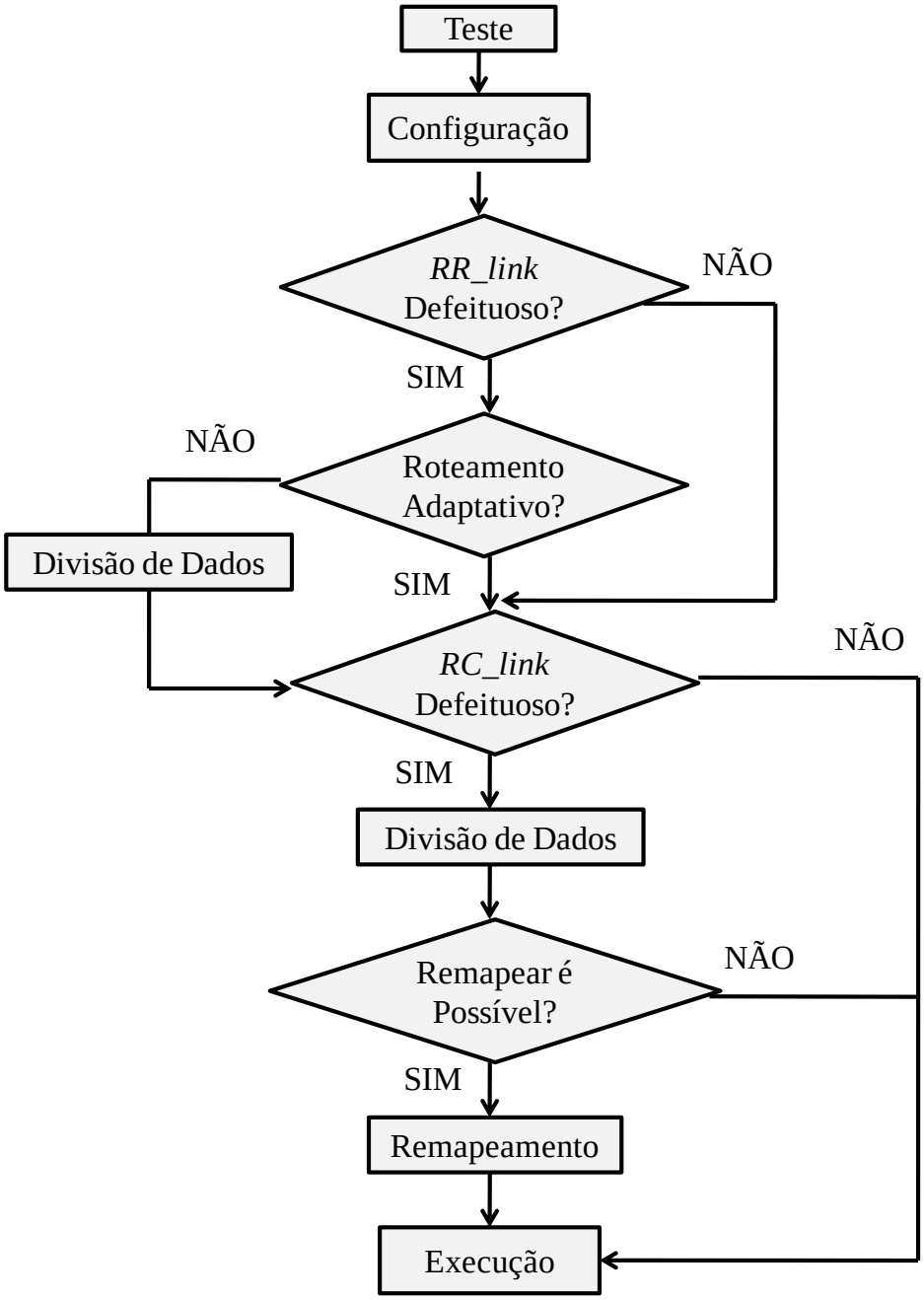


Figura 4.1: Fluxograma das etapas que compreendem o trabalho proposto.

5. ROTEAMENTO ADAPTATIVO

Para realizar a detecção estática e o diagnóstico de falhas nas interconexões os trabalhos de (CONCATTO et al., 2009), (HERVÉ et al., 2009) e (YANG e PAPACHRISTOU, 2009) podem ser considerados. Os dois primeiros trabalhos utilizam a propagação de vetores de teste pela rede para lidar com a detecção e o diagnóstico de falhas nas interconexões, enquanto que o último utiliza teste I_{DDT} (que analisa a variação da corrente dinâmica), testes de atraso e *boudary scan* para detectar defeitos em interconexões de SoCs.

Na estratégia desenvolvida, quando defeitos são detectados nas interconexões através da fase de testes, a primeira solução é evitar as interconexões defeituosas utilizando a técnica de roteamento adaptativo (RA) proposta por (CONCATTO et al., 2009), que é exclusiva para a topologia torus. Esta técnica é definida como consta a seguir:

Em uma rede com duas dimensões e topologia torus MxN, um pacote qualquer pode ter duas rotas possíveis para atingir um mesmo destino: ele pode ir K passos para um caminho (sentido positivo) ou M-K (ou N-K) passos para o caminho oposto (sentido negativo).

Por exemplo, em uma rede 4x3 como mostrada na figura 3.1, um pacote que deixa o roteador R11 para atingir o roteador R12 pode ser transmitido com um único passo para leste (K=1) ou com três passos para oeste (M-K = 4 – 1 = 3), utilizando o link de realimentação proporcionado pela topologia torus. Note que esta característica pode ser utilizada como um recurso para evitar interconexões defeituosas, desde que a informação referente à localização da falha seja armazenada previamente nos roteadores, em registradores específicos.

A fim de lidar com os defeitos de uma maneira mais simples, a técnica desenvolvida por (CONCATTO et al., 2009) tenta utilizar caminhos alternativos para evitar interconexões defeituosas. Para isso, cada roteador é configurado com as informações do teste, e um registrador de 10 bits é necessário para armazenar esta informação em cada roteador. O uso de 10 bits é faz-se necessário para informar se a entrada e a saída de cada canal do roteador possuem defeitos (portas Local, Norte, Sul, Leste e Oeste possuem dois bits cada, um para configurar o estado do canal de entrada e outro para configurar o estado do canal de saída). O bit correspondente a cada porta defeituosa é então configurado para ‘1’, se a falha existe, e o algoritmo de roteamento

verifica estes bits antes de encaminhar o cabeçalho de cada pacote ao próximo roteador. Se o canal de saída escolhido para enviar o dado é indicado como defeituoso, um caminho alternativo é recalculado e reescrito no cabeçalho original do pacote, e o pacote é enviado normalmente através da rota alternativa livre de falhas.

Para o cálculo do novo caminho, cada roteador sabe o tamanho da rede e sua própria posição, e por isso é possível obter o número de passos necessários para o pacote na nova rota. Como consequência, o roteador pode mudar dinamicamente o caminho para alcançar o destino no cabeçalho do pacote quando o endereço original encontra no caminho uma interconexão defeituosa, como mostra a figura 5.1, considerando que o pacote de dados é apresentado na sequência de envio de acordo com o tempo *T(n)*, e que a comunicação dar-se-á do roteador A para o roteador C, evitando, assim, a interconexão defeituosa entre os roteadores B e C.

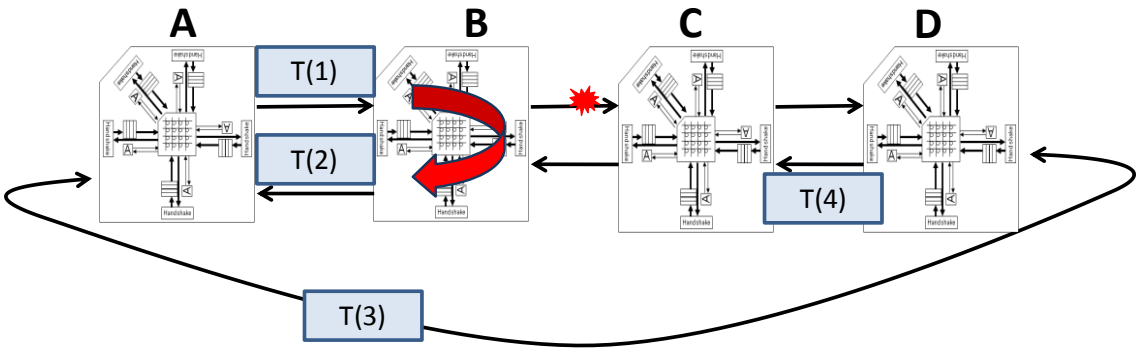


Figura 5.1: Exemplo de uma situação em que o cabeçalho do pacote é alterado para mudar a rota original, a fim de evitar um caminho defeituoso.

Como exemplo, também se pode considerar a rede apresentada na figura 5.2 para o MPEG4. Um pacote foi inserido na rede através do núcleo *MED CPU* (*R4*) e, para exemplificar, tem por objetivo atingir o destino RAST (*R3*). Porém, ao invés de utilizar diretamente a interconexão entre *R6-para-R3*, o cabeçalho do pacote será alterado no roteador *R6*. O caminho tradicional da comunicação acontece através das interconexões destacadas na seguinte ordem: *MEDCPU-para-R4*, *R4-para-R6*, *R6-para-R3 e R3-para-RAST*, mas após atingir a coluna destino, o caminho que seria de *R6-para-R3* não pode ser utilizado, pois existe uma falha no caminho. Então, o caminho precisa ser alterado da seguinte forma: *R6-para-R9*, depois *R9-para-R12* e então a interconexão de realimentação da rede torus deve ser utilizada através de *R12-para-R3* a fim de alcançar o núcleo destino. A parte da comunicação que é alterada em tempo de execução consiste em todo o caminho realizado na vertical. Embora cada flit da mensagem seja roteado novamente, o impacto global na comunicação e na potência é normalmente baixo, quase imperceptível, pois o caminho alternativo não é tão maior que o caminho original (na maioria dos casos analisados). A área extra para a implementação deste trabalho de (CONCATTO et al., 2009) é também consideravelmente pequena, em torno de 1%, apenas por inserir uma verificação extra

que visa recalcular o caminho de um pacote na presença de falha no caminho original estabelecido.

O roteamento adaptativo pode lidar apenas com falhas entre dois roteadores (*RR_links*) desde que não exista mais de uma falha no caminho (a fim de evitar *deadlock*). O máximo de falhas que a técnica de roteamento adaptativo permite tolerar facilmente é de até uma falha por cada "linha de interconexões" e uma falha por cada "coluna de interconexões", a fim de garantir que situações de *deadlock* não irão surgir. Em uma rede 3x4, Uma "linha de interconexões" é considerada, neste trabalho, como todas as interconexões que ligam os roteadores R1, R2 e R3 na horizontal, como se pode observar na figura 5.2, e uma "coluna de interconexões" corresponde a todas interconexões que ligam os roteadores R1, R4, R7 e R10 na vertical, por exemplo. Então, o roteamento adaptativo é considerado livre de *deadlock* sempre que ocorrer apenas 1 falha em cada grupo de interconexões na horizontal ou vertical de cada coluna ou linha. Isso significa que numa rede 4x3, por exemplo, o máximo de falhas que podem ser toleradas pelo RA com facilidade são 4 falhas na horizontal, considerando que cada uma delas acontece em um linha distinta, e 3 falhas na vertical, considerando que cada uma delas ocorre em uma coluna diferente.

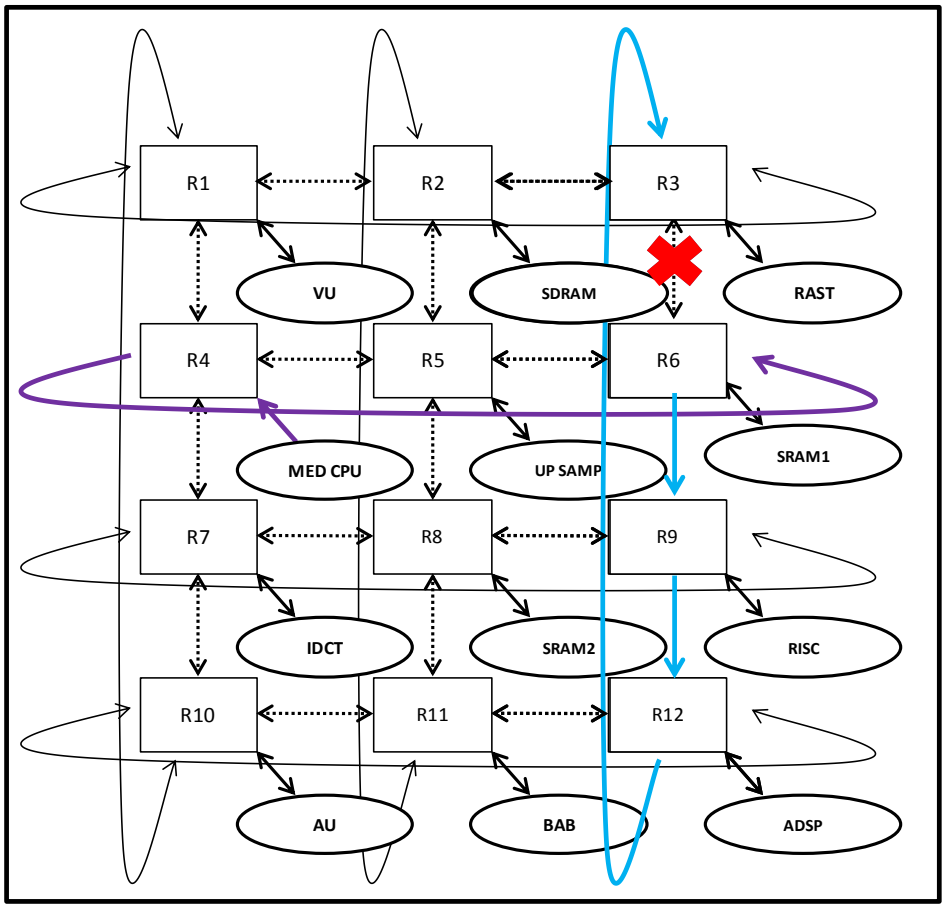


Figura 5.2: Exemplo de roteamento adaptativo para um caso de uma falha entre a interconexão do roteador 3 e do roteador 6.

Contudo, a estratégia proposta por (CONCATTO et al., 2009) não pode lidar com situações em que ambas as entradas ou saídas de um roteador, em uma mesma direção, encontram-se defeituosas, como já apresentado na figura 3.9 e novamente apresentado na figura 5.3. Considerando um exemplo de uma falha que afeta a entrada leste de um roteador e outra falha que afeta a entrada oeste deste mesmo roteador (R11), elas não irão permitir que ele receba dados através do eixo X de comunicação da rede. Essa situação pode acontecer também bloqueando o eixo Y (R6). Além deste problema, o trabalho de (CONCATTO et al., 2009) não pode lidar com falhas nas interconexões que comunicam núcleo (MED CPU e IDCT) com roteador (R2 e R3, respectivamente), reduzindo consideravelmente a eficiência do trabalho quando estas falhas são consideradas, já que estas interconexões normalmente representam no mínimo 20% de qualquer rede (considerando que cada roteador tem 5 portas e uma delas sempre será a porta local que conecta-se a um núcleo). Para ambos os casos não solucionados por (CONCATTO et al., 2009) é que a técnica a seguir foi desenvolvida, a fim de ser utilizada somente para casos que não podem ser solucionados pela técnica de roteamento adaptativo.

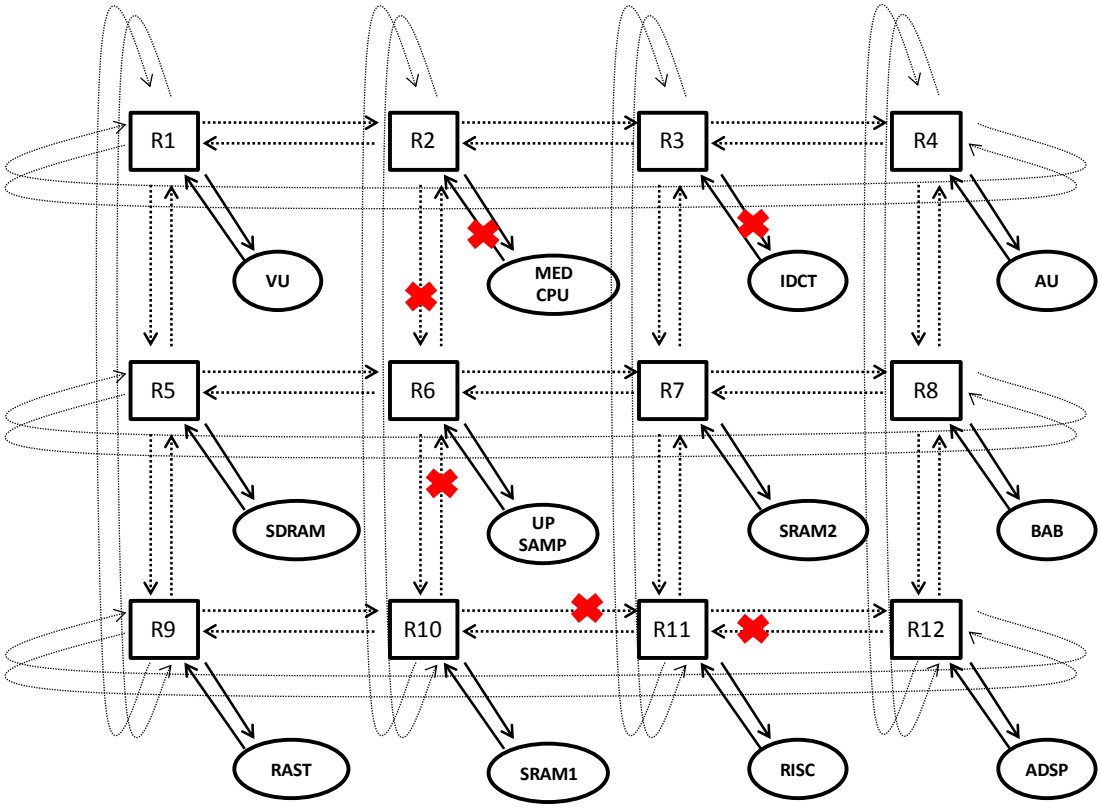


Figura 5.3: Exemplos de situações de falhas que o roteamento adaptativo não pode lidar.

6. DIVISÃO DE DADOS NOS CANAIS DA REDE INTRA-CHIP

Quando nenhum caminho alternativo é possível através dos recursos oferecidos pelo uso da técnica de roteamento adaptativo, então é necessário que exista uma estratégia alternativa para prover a comunicação da rede, a fim de evitar áreas de isolamento devido às interconexões defeituosas. Sabendo-se que uma interconexão está defeituosa, é improvável que todos os fios desta interconexão estejam realmente com falhas, e por isso a técnica proposta visa permitir que apenas os fios livres de falhas de uma interconexão sejam utilizados, provendo sempre uma comunicação limitada, com a metade da capacidade original da interconexão.

Embora a taxa de comunicação do canal seja reduzida, uma vez que ele está operando com a metade da capacidade, a vantagem desta proposta é não deixar que um fio defeituoso torne uma interconexão completamente inutilizável para a rede, apenas por causa do fio que não pode ser utilizado corretamente. Sendo assim, os dados que antes trafegavam em canais com uma largura de X bits, irão ser divididos no momento em que atingem a interconexão defeituosa, e dali seguem o percurso ocupando apenas X/2 bits da interconexão até encontrarem o alvo final, onde através da interface de rede são reagrupados novamente. Desta forma, quando os dados são divididos para ocupar um canal defeituoso, eles utilizam apenas os fios considerados bons naquela interconexão, através do uso de multiplexadores que permitem a seleção de quais fios serão utilizados. A princípio esta técnica dobra o tempo de comunicação para todos aqueles pacotes que passam pelo caminho com falhas. Porém, esta técnica é utilizada somente para os casos não solucionados pelo roteamento adaptativo, tornando-a vantajosa na maioria das vezes, já que evita isolar a comunicação de roteadores e núcleos que possuem uma ou mais falhas nas interconexões ao redor. Além disso, quando nenhuma falha é diagnosticada, os recursos podem ser desligados a fim de reduzir potência e melhorar o tempo de comunicação entre os núcleos, já que a frequência do circuito pode ser melhor há menos hardware no caminho crítico, ou seja, quando a divisão de dados (DD) não é utilizada. De qualquer forma, por exemplo, quando uma interconexão necessita da divisão de dados, apenas os roteadores envolvidos na comunicação entre a interconexão defeituosa precisam estar habilitados

para trabalhar com a técnica, o que significa que o acréscimo na potência e na energia pode ser razoavelmente pequeno quando poucas falhas existem na rede.

Quando a divisão de dados é utilizada, um pacote que passa pela interconexão defeituosa leva o dobro do tempo para atingir o destino, já que o dado é dividido e transmitido pela metade, necessitando do dobro de ciclos. Para permitir que cada parte do dado seja transmitida corretamente, então apenas os fios que não possuem falhas são utilizados. Isso é possível porque multiplexadores estão posicionados na entrada e na saída de cada canal do roteador, permitindo escolher apenas os fios livres de falhas para serem utilizados na transmissão, como mostra a figura 6.1. O controle dos multiplexadores é configurado de acordo com os resultados obtidos previamente em tempo de teste, e varia de acordo com o tamanho do multiplexador necessário (ou seja, varia com a quantidade de fios de cada interconexão).

Teoricamente, os fios que são constatados como defeituosos e/ou descartados na comunicação poderiam ser configurados para receber os valores de *ground* ou *vdd*, de acordo com o tipo de falha diagnosticada, a fim de minimizar as chances de qualquer falha influenciar sobre o valor dos outros fios que atuam na comunicação. Sendo assim, os valores de *ground* ou *vdd* poderiam "neutralizar" o impacto da falha em outros fios. Para este trabalho, a fim de simplificar o hardware e minimizar o impacto do tempo de comunicação no caminho crítico, será considerado apenas o uso de 50% das interconexões que estejam boas. Isto significa que se apenas um fio estiver defeituoso, 50% da interconexão não será utilizada (incluindo o fio defeituoso), o que obviamente implica em também não utilizar algumas interconexões em boas condições. Mesmo que apenas 1 fio esteja defeituoso em cada interconexão, a utilização dela fica estabelecida em 50% da sua própria capacidade. Sendo assim, fios que poderiam sofrer qualquer efeito influenciado pelo valor que consta no fio defeituoso (por exemplo, situações de *crosstalk*), também devem ser descartados, já que a fim de simplificar o hardware utilizado se optou por não utilizar valores de *ground* ou *vdd* nos fios descartados.

A abordagem de usar divisão de dados consiste em proteger até 50% das interconexões e permite que cada bit seja transportado em fios com até 50% de deslocamento, como mostrado na figura 6.1. Por exemplo, um fio que se encontra na primeira posição, da primeira metade de uma interconexão com X fios, pode ser transportado na posição 1 + X/2 a fim de evitar falhas que possam estar na primeira metade da interconexão. A mesma idéia vale para fios defeituosos que estão na segunda metade da interconexão, como é o caso do bit localizado na posição X/2+1, que pode ser deslocado para o transporte em até X/2 posições acima (posição 1). Para resumir a idéia citada anteriormente, pode-se dizer que a metade de baixo da interconexão pode ser deslocada em até 50 % dos fios acima, e a metade de cima pode ser deslocada em até 50% dos fios abaixo.

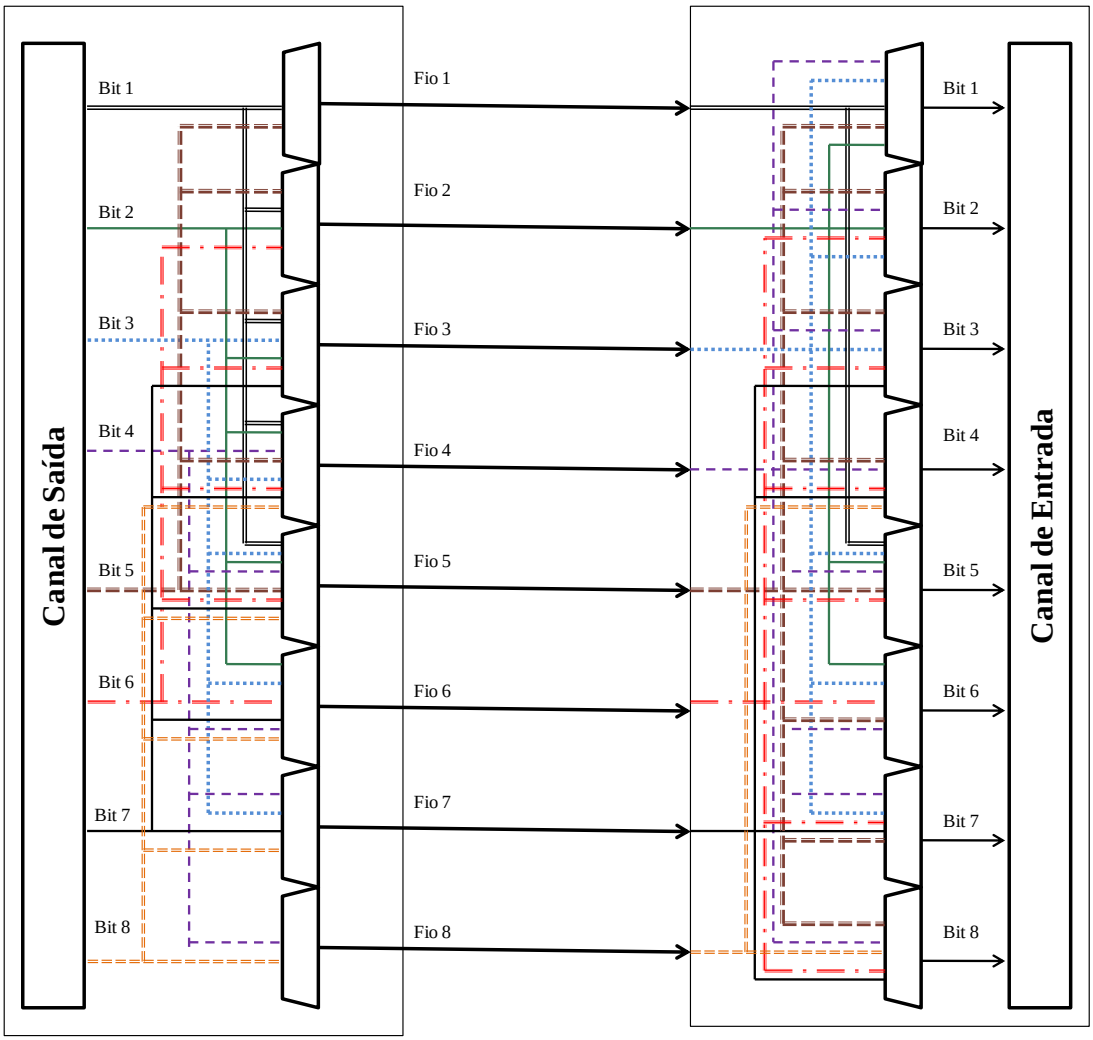


Figura 6.1: Posicionamento dos multiplexadores para os fios de uma interconexão com 8 bits. Apenas fios bons podem ser selecionados para a transmissão dos dados.

Para os fios que se encontram no meio da interconexão, o número de entradas nos multiplexadores é igual ao número de fios que a interconexão possui. Sendo assim, é necessário considerar o pior atraso de 2 multiplexadores 8:1 no caminho crítico (um multiplexador no canal de saída, e outro no canal de entrada) para esta solução que considera canal de 8 bits. Quando consideramos apenas os fios mais ou menos significativos, o número de entradas é reduzido considerando que o primeiro e o último bit não possuem fios localizados antes e/ou depois da sua própria posição, respectivamente. Para interconexões maiores de 16 ou 32 bits, o atraso dos multiplexadores aumenta consideravelmente, e por isso a proposta pode ser reavaliada para tolerar um número menor de falhas, como por exemplo, apenas 25% de falhas nas interconexões ao invés de 50%, permitindo que o número de entrada nos multiplexadores possa ser reduzido de acordo com a tolerância desejada.

Para contornar o hardware que fragmenta a informação em duas partes, quando ele não é necessário, pode-se utilizar mais um multiplexador em cada interconexão, como mostra a figura 6.2. Nela, os blocos da divisão de dados (DD) são compostos pela estrutura apresentada na figura 6.1, e a divisão de dados varia de acordo com o número de bits da interconexão (para esta abordagem apenas 8 bits são considerados). O sinal *Sel_DD* que seleciona os multiplexadores é responsável por selecionar o dado proveniente do bloco DD ou da interconexão e é configurado previamente com os resultados obtidos na etapa de testes.

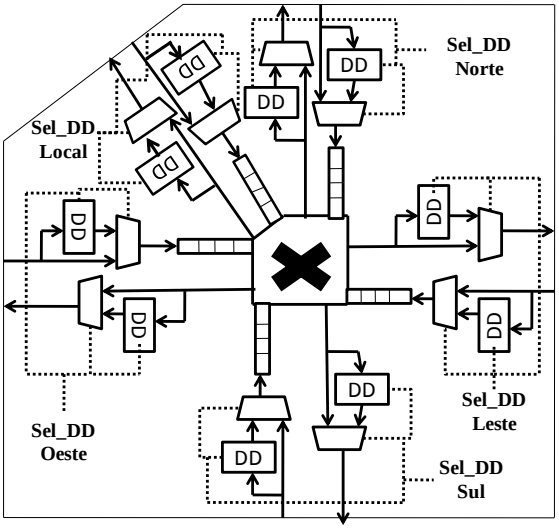


Figura 6.2: Diagrama de blocos do roteador com divisão de dados.

O dado proveniente de DD é escolhido quando a divisão de dados encontra-se ativa no roteador, ou em caso contrário (na ausência de falhas), o dado escolhido vem diretamente da interconexão. Baseado nos testes de manufatura ou testes funcionais, os fios livres de falha são escolhidos e esta informação pode ser configurada através de entradas externas ou de uma cadeia *scan* conectada aos registradores de configuração. Quando não existem falhas em torno de um roteador, o bloco DD pode ser desligado através da técnica de *sleep transistor* para reduzir a potência do circuito (LONG e HE, 2004) (SHI e HOWARD, 2006).

Quando a divisão de dados é utilizada, todos os flits são divididos em 2 partes com o mesmo tamanho, e isso significa que o cabeçalho com as informações do roteamento também precisa ser dividido. Cada cabeçalho é identificado com valores específicos de *begin-of-packet* e *end-of-packet* como consta na seção 2.3. Então, quando existe uma interconexão defeituosa e o roteador está utilizando a divisão de dados no canal de entrada, o cabeçalho precisa ser devidamente identificado a fim de indicar ao roteador que ele está fragmentado em duas partes, e que todo o restante do pacote também está fragmentado. Assim, os sinais de *bop* e *eop* são codificados de forma especifica para "11", indicando que todos os pacotes de dados que chegam por aquele caminho estão divididos em duas partes. Isto é importante para que o roteador identifique a informação corretamente.

Somente quando os flits atingem o destino final, nós consideramos que os dados são agrupados novamente na interface de rede, permitindo que cada metade da informação possa viajar na rede por mais de uma interconexão com defeitos sem prejudicar ainda mais a latência da informação transmitida, uma vez que ela já se encontra dividida. Apenas o cabeçalho precisa ser agrupado para que o endereço de destino seja obtido. Se existe mais de uma falha no caminho do dado que requer o uso de DD, a latência de um ciclo é inserida na rede cada vez que o cabeçalho precisar ser novamente dividido ou reagrupado. Sempre que um cabeçalho chegar dividido do outro lado da interconexão, um ciclo de latência é inserido por esperar que a segunda metade do cabeçalho chegue, a fim de identificar corretamente o caminho.

Após cada transmissão através do fio defeituoso, a primeira metade dos dados é reagrupada com o uso do bloco DD nos fios posicionados na primeira metade da interconexão, bem como a segunda metade dos dados é reagrupada na segunda metade da interconexão, de acordo com a ordem de envio, e assim a transmissão dos dados na rede continua até que eles sejam reagrupados definitivamente na interface de rede, antes de atingirem o núcleo. Um exemplo de divisão de dados é mostrado a seguir na figura 6.3, em que os multiplexadores foram omitidos pelo bloco DD da figura a fim de simplificar a visualização do método utilizado.

Para considerar uma situação onde a divisão de dados é necessária, vamos levar em conta uma variação da figura 5.2, em que uma falha entre um núcleo e um roteador foi inserida além da falha entre os roteadores R3 e R6. Neste caso, considera-se a comunicação de MED CPU para RAST, como está apresentado na figura 6.4, e não existe outro caminho alternativo para que o RA seja utilizado entre R3 e RAST, e então o bloco DD precisa estar ativo para solucionar o caso. Como apenas o roteador R3 irá utilizar a divisão de dados, todos os outros roteadores podem ter o bloco DD desligado a fim de reduzir a potência e a energia da rede. Para um roteador que agrega RA e DD foi definida a sigla RRADD (roteador com roteamento adaptativo e divisão de dados).

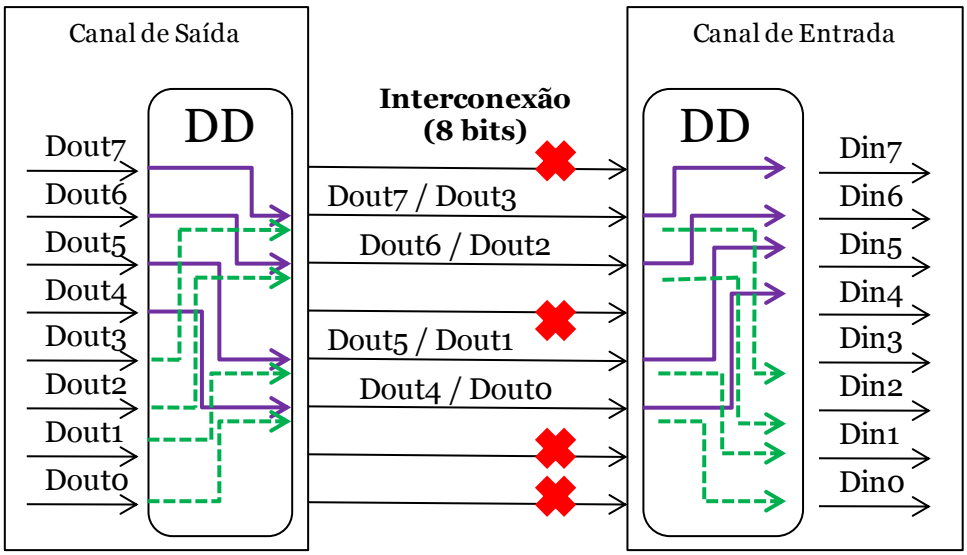


Figura 6.3: Exemplo do uso da divisão de dados em uma interconexão de 8 bits com 50% de fios defeituosos.

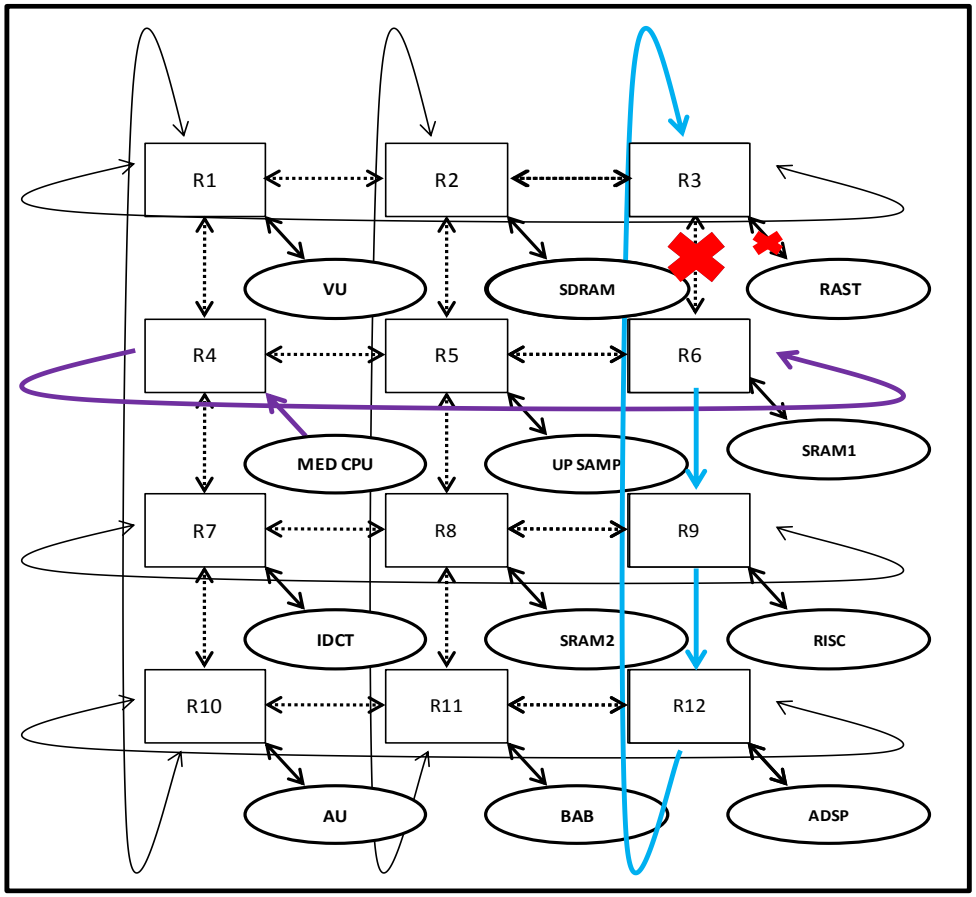


Figura 6.4: Situação que exige a combinação de RA e DD para tolerar as falhas da rede.

É muito importante ressaltar que a técnica de divisão de dados pode causar um impacto variável no tempo de comunicação, que pode ser de até 2 vezes o tempo original numa situação sem falhas, para o pior caso. Porém, este impacto não é regra, como mostram algumas situações de DD ilustradas na figura 6.5 e de acordo com a taxa de envio estabelecida para os pacotes. Para as situações em que o envio de pacotes tem uma defasagem entre um envio e outro de no mínimo o intervalo que corresponde ao envio de um pacote, o impacto da divisão de dados é mínimo, sendo de apenas 4 ciclos para o caso em que o tamanho de pacote é estabelecido em 4 flits. Para as situações em que a taxa de envio é a máxima permitida, de 4 flits a cada 4 ciclos, é que a divisão de dados pode causar uma penalidade maior. Desta forma, não se pode generalizar o uso da divisão de dados como uma técnica que sempre irá influenciar significativamente o tempo de comunicação, mas sim como uma técnica de influência variável em acordo com as taxas de envio que são utilizadas na interconexão defeituosa.

Nas situações em que o remapemanento é combinado com DD, procura-se sempre associar a interconexão defeituosa com o local onde as taxas de envio são mais similares a situação 1 da figura 6.5, a fim de minimizar o impacto na comunicação, ocupando espaços de tempo que estariam ociosos na rede para transmitir os dados divididos pelo bloco DD. O trabalho aqui apresentado difere de (BRAGA et al., 2010)

porque não utiliza mecanismo de retransmissão. Quando BRAGA tem 1 falha permanente em cada metade da interconexão, nem mesmo a retransmissão poderá solucionar, enquanto que nossa proposta utiliza apenas os fios detectados como livres de falhas para realizar a transmissão em duas etapas.

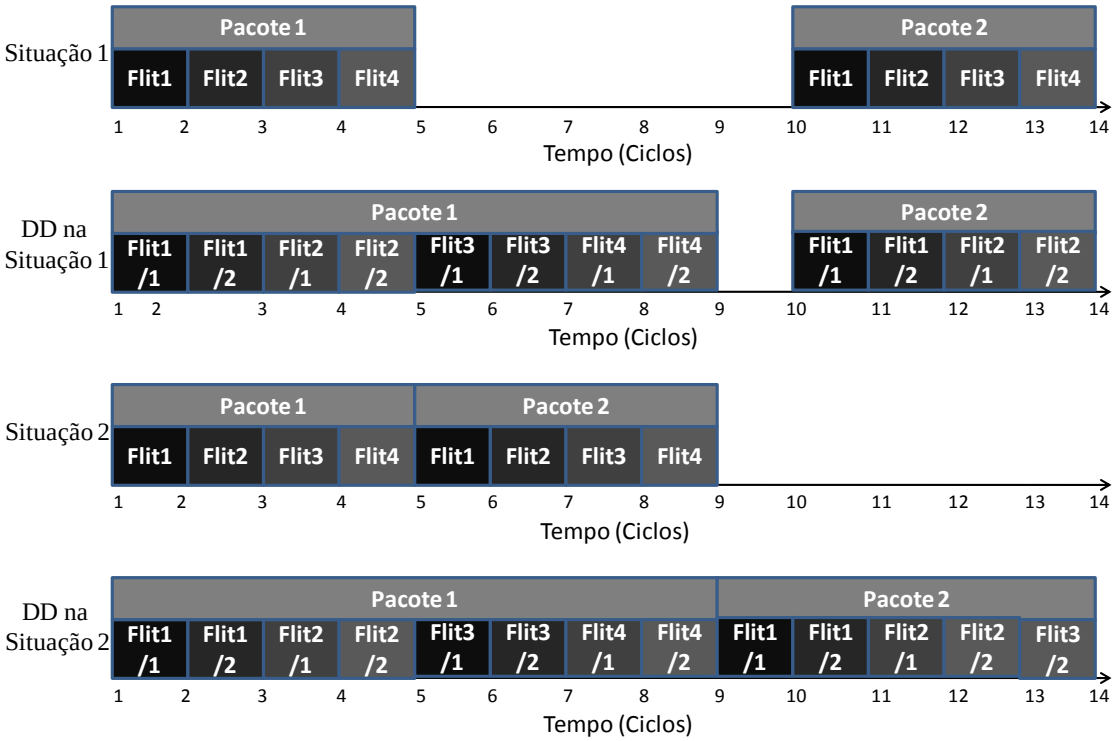


Figura 6.5: Exemplos de impacto no tempo de comunicação com a utilização de DD.

Em raras ocasiões em que as falhas atingem mais de 50% de uma interconexão, e o roteamento adaptativo também não pode ser uma solução válida nem quando o remapeamento for aplicável, é então necessário que a interconexão defeituosa seja evitada. Se não for possível evitá-la, o que consequentemente causaria regiões de isolamento, provavelmente o chip deverá ser descartado.

Em resumo, por utilizar o método proposto, quando múltiplos defeitos são detectados nas interconexões, os passos para manipular uma melhor utilização do chip são: aplicar roteamento adaptativo ou então utilizar divisão de dados, e utilizar o remapeamento quando a falha tem muita influência na comunicação, como será apresentado na próxima seção. Se muitas falhas são encontradas nas interconexões do circuito, é então necessário utilizar uma combinação das técnicas propostas, e de acordo com a quantidade e a localização das falhas, uma solução com melhores resultados entre as técnicas disponíveis pode ser configurada.

6.1 Divisão de Dados Otimizada

A divisão de dados pode ser otimizada com base nas previsões feitas por (DEHON e NAEIMI, 2005). Como já dito anteriormente, as previsões afirmam que a taxa de fios e conexões defeituosas pode ser de até 15%. Com base nessa informação, objetivou-se reduzir o número de ligações entre os multiplexadores e os fios dentro do bloco DD. Assim, a área foi reduzida e o projeto tornou-se mais escalável para uma largura de dados com 32 bits nas interconexões.

Com a realização da otimização, somente será possível deslocar um dado da primeira metade da interconexão até três vezes abaixo, e um dado da segunda metade até três vezes acima. Essa condição permite que a quantidade de multiplexadores 2:1 utilizados em cascata seja reduzida até o comportamento de um multiplexador 8:1 para os fios do meio da interconexão, considerando uma interconexão de 32 bits. Os fios do meio da interconexão podem ser considerados os mais críticos, uma vez que eles podem transmitir dados tanto da parte mais e menos significativa do dado.

Com a nova abordagem proposta, a quantidade de falhas toleradas em cada metade da interconexão chega a 18,75%, pois até três fios defeituosos podem ser tolerados nos primeiros e nos últimos 16 bits de uma interconexão com largura de 32 bits.

Outra melhoria que foi também realizada consiste na inserção de um elemento de memória denominado *latch* para armazenar o dado transmitido em cada metade da interconexão. Com isso, o controle do chaveamento dos multiplexadores foi programado para ser sensível ao nível, e desta forma há duas transmissões de dados dentro do período de um ciclo de relógio. A figura 6.6 exibe onde os elementos de memória sensíveis ao nível foram adicionados. L1 armazena dados transmitidos na primeira fase do relógio (nível alto) e L2 armazena a segunda metade do flit transmitido (nível baixo), uma vez que apenas 50% da interconexão é utilizada em cada transmissão quando o bloco DD é requerido. Para entender o novo mecanismo de transmissão, pode-se observar na figura 6.7 o diagrama de dados para a transmissão de flits, considerando uma representação hexadecimal. O que se pode perceber é que para o dado ser transmitido da fonte para o destino foi necessário um ciclo a mais de latência, inserido pela utilização dos *latches*. Porém, na abordagem anterior, cada comunicação que utilizava a divisão de dados levaria no mínimo o dobro de ciclos para efetuar a comunicação. Essa melhoria só foi possível de ser realizada porque a área do bloco DD foi limitada em ligações de até oito entradas nos multiplexadores para cada interconexão, e assim foi possível inserir *latches* na estrutura sem uma considerável penalidade.

Para garantir a correta operabilidade dos *latches* com a transmissão de dados ocorrendo em duas fases, foi necessário constatar que os fios da rede podem trabalhar até duas vezes mais rápidos que a freqüência permitida pelo roteador. Os resultados que comprovam esta afirmação serão apresentados na seção 8, juntamente com os resultados

da nova abordagem como área, frequência, potência e energia para a tecnologia de 65 nm.

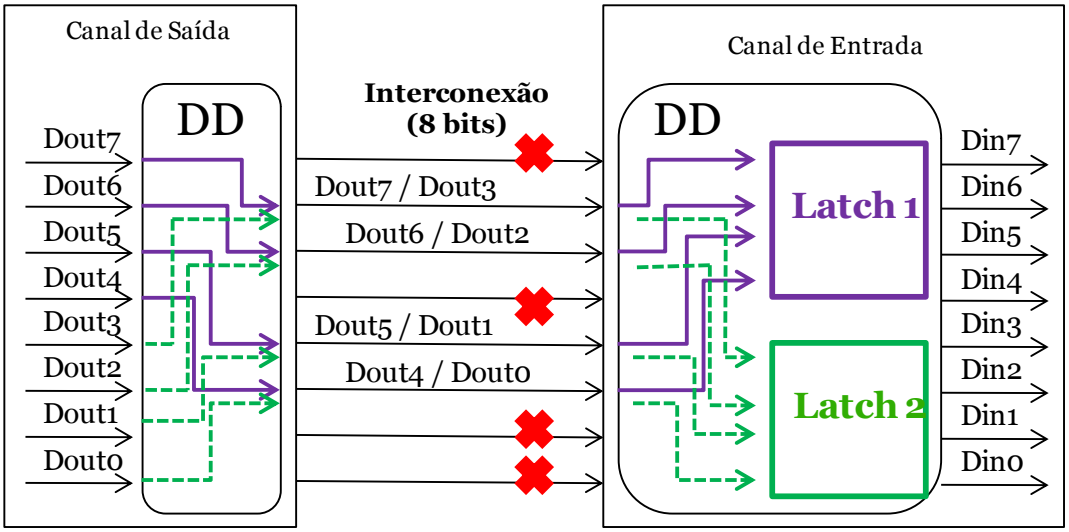


Figura 6.6: Dois *latches* foram inseridos após a redução do bloco DD para minimizar o impacto na comunicação.

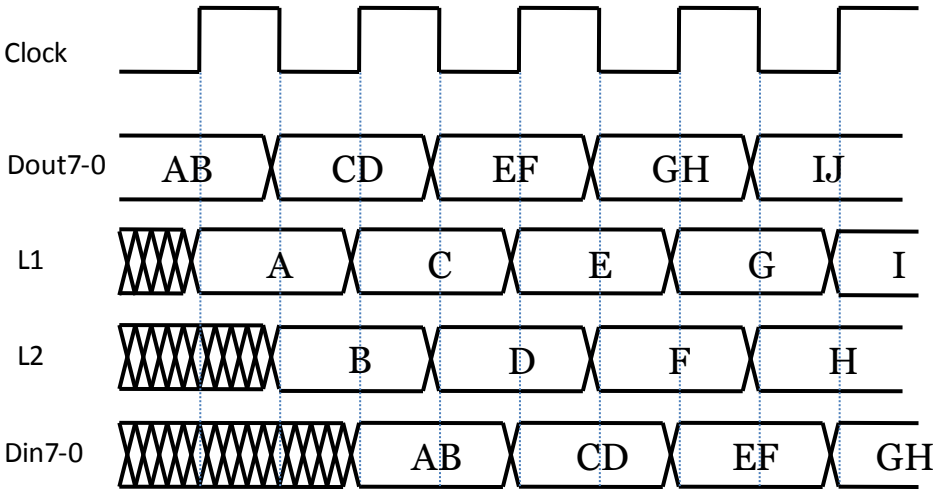


Figura 6.7: Diagrama de dados em uma interconexão que utiliza DD com *latches*.

7. REMAPEAMENTO DE TAREFAS

O remapeamento de tarefas pode ser facilmente utilizado com base nos resultados do diagnóstico das falhas. Uma vez detectadas as falhas, núcleos com baixa comunicação podem ser alocados ao redor das falhas a fim de minimizar o impacto da falha na comunicação, sem grandes penalizações para a comunicação da rede. Para redes compostas somente por núcleos homogêneos, o remapeamento é simples, uma vez que todos os núcleos possuem a mesma funcionalidade e área, tornando a alocação das tarefas mais simples. Quando os núcleos são heterogêneos, a solução pode ser factível se houver núcleos com a mesma funcionalidade disponíveis para o remapeamento das tarefas, como, por exemplo, regiões redundantes na rede, memórias idênticas, 2 processadores iguais, etc... Além disso, a rede intra-chip pode ser acoplada aos núcleos em um leiaute separado, como é o caso dos circuitos em 3 dimensões (BANERJEE et al., 2011). Assim, o projeto da disposição dos núcleos pode ser realizado com arranjos diferentes, e de acordo com a localização das falhas nas interconexões da rede intra-chip, escolhe-se o arranjo com menos impacto na comunicação. Se o remapeamento não for uma opção, então apenas as soluções de roteamento adaptativo e divisão de dados são empregadas na rede.

Por simplicidade, o remapeamento foi escolhido para ser a última abordagem utilizada, conforme mostra o fluxo da figura 4.1. Porém, ele poderia também ser adotado como a primeira estratégia para a tolerância de falhas, alterando parcialmente o fluxo da figura 4.1 para remapeamento, roteamento adaptativo e divisão de dados, respectivamente. O remapeamento é a técnica que pode oferecer mínimo impacto da falha na rede, já que pode mapear a falha para interconexões não utilizadas ou com pouco tráfego, de acordo com a aplicação escolhida.

5.1 Remapeamento de Redes com Núcleos Homogêneos

Para explicar como ocorre o remapeamento de tarefas em redes com núcleos homogêneos, é apresentado na figura 7.1 um grafo qualquer de uma comunicação hipotética, apenas para exemplificação, com 4 núcleos se comunicando em diferentes taxas.

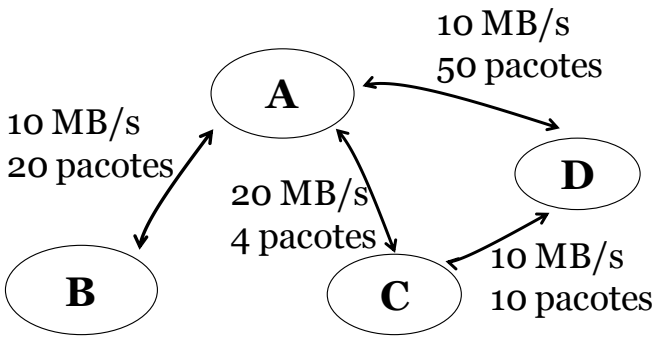


Figura 7.1: Grafo de comunicação para um caso hipotético.

De acordo com a figura 7.1, quando nenhuma falha existe na rede, o melhor mapeamento pode ser obtido baseado nos resultados estimados de atraso que cada comunicação possui. Comunicações muito intensas tendem a ser posicionadas próximas, e núcleos com baixas taxas de comunicação podem normalmente estar situados distantes um do outro, já que a comunicação deles tende a ter pouca influência na rede. Para o grafo de comunicação apresentado na figura 7.1, o respectivo mapeamento escolhido consta na figura 7.2.

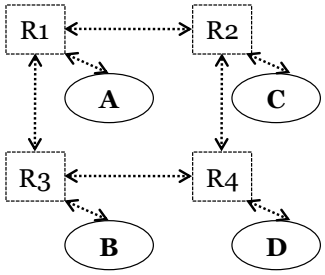


Figura 7.2: Mapeamento escolhido para o grafo da comunicação anterior.

Para obter os resultados de atraso para cada núcleo, a equação abaixo é utilizada, onde *#_pacotes* representa o número estimado de pacotes a serem enviados por cada núcleo, *tamanho_do_pacote* é dado de acordo com a largura do canal, em bits, e a *taxa* de envio dos pacotes para cada núcleo, que é medida em Mega Bytes por segundo:

$$Atraso = (\#\_pacotes * tamanho\_do\_pacote) / taxa$$

Com os resultados alcançados através da equação acima, pode-se obter o gráfico da figura 7.3, que mostra o tempo estimado de comunicação de cada núcleo da rede, considerando os *CR_links*. Para aqueles núcleos que possuem mais de uma comunicação ativa, os tempos foram somados, e para os resultados apresentados neste capítulo a largura de canal foi desconsiderada para o cálculo do tempo de comunicação, uma vez que todos os núcleos possuem a mesma largura de canal (8 bits).

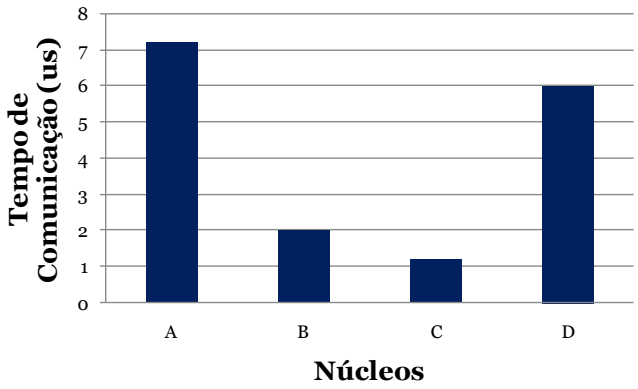


Figura 7.3: Tempo de comunicação para cada núcleo da rede no mapeamento escolhido.

Como se pode ver na figura 7.3, os núcleos A e D possuem o maior tempo de comunicação na rede. Então, em caso de falhas em algum *CR_link*, é desejável que esta falha não esteja no caminho de comunicação dos núcleos A e D, uma vez que o impacto no tempo de comunicação total pode se tornar altamente prejudicial para a rede considerando que ambos são núcleos com um elevado tempo de comunicação.

Se uma falha é detectada como consta na figura 7.4, a melhor solução para reduzir o impacto no tempo de comunicação é fazendo-se uma simples variação no mapeamento original escolhido, para minimizar o impacto na comunicação quando existe a presença de falhas na rede.

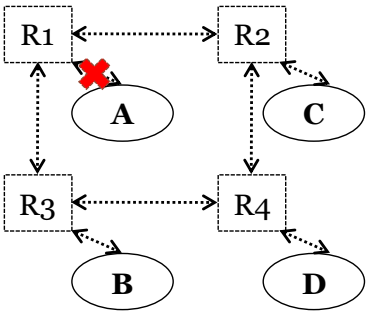


Figura 7.4: Situação crítica em que é detectada uma falha no *CR_link* do núcleo A.

A solução utilizada é baseada no espelhamento do mapeamento original escolhido inicialmente: ou seja, o mapeamento é alterado para a forma como ele é visto em um espelho, pois esta é a maneira mais fácil de trocar os elementos da rede sem mudar a verdadeira orientação definida como mapeamento original. Para manter as características do mapeamento escolhido originalmente em tempo de projeto, optou-se por apenas espelhar a disposição dos núcleos da rede, sendo o espelhando dos núcleos na posição vertical, horizontal ou em ambas as direções ao mesmo tempo. Assim, sempre é possível obter até 4 arranjos de um mapeamento, considerando um número qualquer de núcleos em uma rede com topologia grelha ou torus. No caso do exemplo escolhido, todos os 4 núcleos podem ser mapeados com diferentes arranjos para a posição defeituosa, de forma muito simplificada, como é mostrado na figura 7.5, onde

as 4 variações são apresentadas. As 4 possibilidade de mapeamento apresentadas não afetam a organização original definida para o mapeamento, e por isso facilitam a reoganização dos núcleos.

A técnica de utilizar um simples espelhamento pode ser tranquilamente aplicada para redes com um maior número de núcleos, desde que a rede utilize a topologia grelha ou torus. Existem algumas situações em que alguns núcleos não podem ser alterados, como, por exemplo, em uma rede 3x3, onde o núcleo do centro não possui outro em posição similar para ser substituído apenas através do espelhamento. Para estas situações não é possível diminuir o impacto da falha na comunicação, já que o núcleo não pode ser alterado. Para qualquer arranjo de núcleos na topologia grelha ou torus, sempre irá existir no máximo 4 combinações possíveis de um mesmo mapeamento, sem qualquer influência no tempo de comunicação quando não existem interconexões defeituosas na rede, bastando organizar a rede em diferentes orientações: original, espelhada vertical, espelhada horizontal, ou com ambos esplhamentos.

Para o exemplo apresentado neste capítulo, calculou-se o tempo de comunicação para cada núcleo quando ele foi mapeado para a posição com falha no *CR_link*, como pode-se observar na figura7.6.

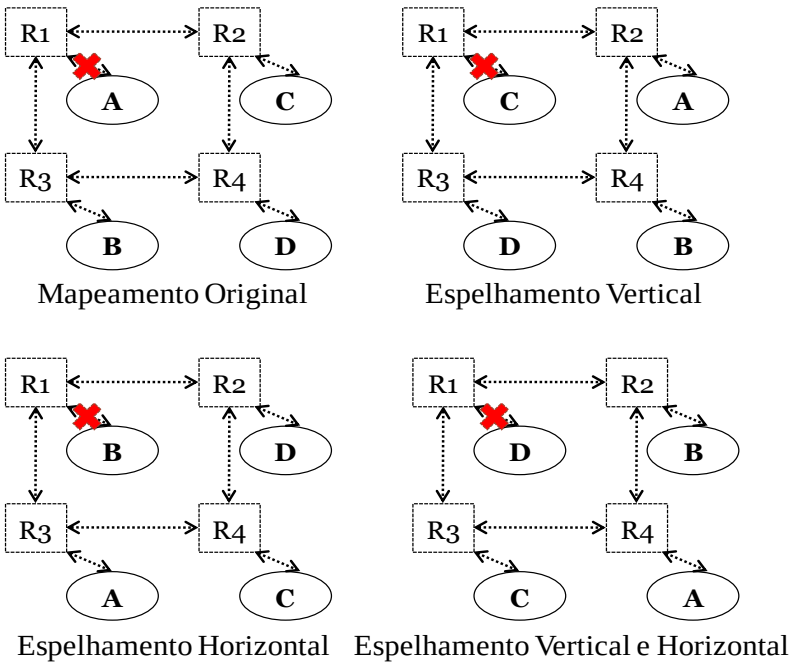


Figura 7.5: Possibilidades de mapeamento adotadas com o espelhamento dos núcleos.

Quando uma falha afeta um *CR_link*, a técnica utilizada para tolerar o defeito é dividir o dado e enviá-lo fragmentado em 2 partes pela interconexão defeituosa, necessitando, em uma primeira abordagem, o dobro de ciclos para as comunicações que passam pelo local defeituoso. Isso significa que mapeando o núcleo C para a posição próxima da falha, o impacto no tempo de comunicação pode ser minimizado quando comparado com os outros núcleos mapeados para a mesma situação, pois dobrando-se o

tempo de comunicação do núcleo C temos o menor impacto se compararmos com as outras possibilidades.

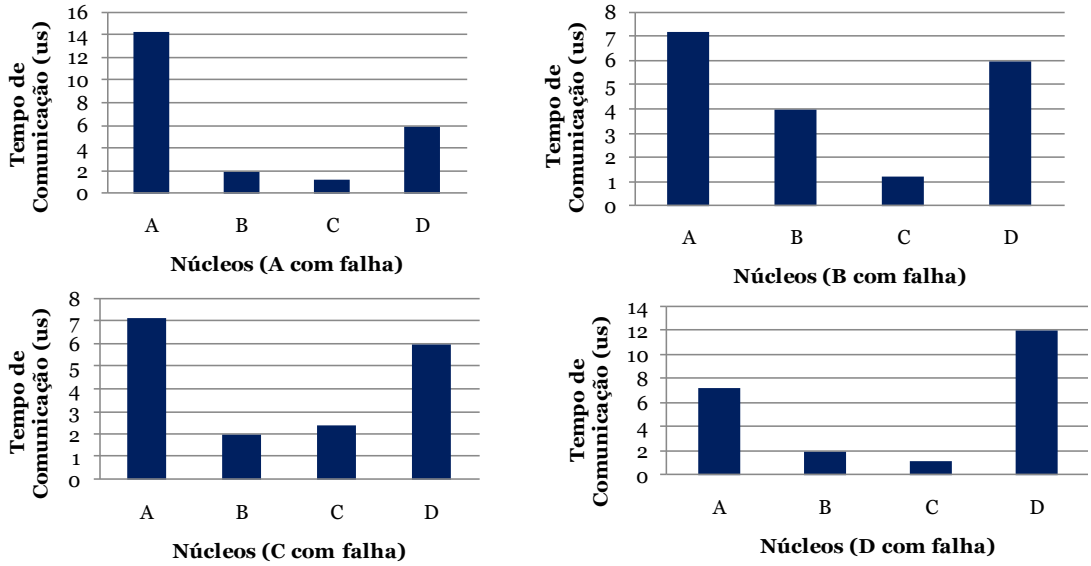


Figura 7.6: Tempo de comunicação para cada núcleo quando mapeado no *CR_link* defeituoso de acordo com a figura 7.5.

Quando comparamos os tempos de comunicação na presença de falhas, na tabela 7.1, o que se pode perceber é que o núcleo com menor impacto no tempo de comunicação é o núcleo C, porque a soma de todos os tempos de comunicação será menor para a situação em que a falha afeta o núcleo C.

Tabela 7.1: Tempo de comunicação total para cada situação de mapeamento.

Situação	**Mapeamento**	**Soma Total do Tempo Comunicação (μs)**
Sem Falhas	Original	16,4
Núcleo A na posição defeituosa	Original	23,6
Núcleo B na posição defeituosa	Espelhamento horizontal	18,4
Núcleo C na posição defeituosa	Espelhamento vertical	17,6
Núcleo D na posição defeituosa	Espelhamento horizontal e vertical	22,4

O impacto no tempo de comunicação é calculado com base no fato de que o *CR_link* não pode ser descartado na comunicação (pois não existe um caminho alternativo a ele), e assume-se que para mandar cada dado através de uma interconexão defeituosa é necessário em geral o dobro de ciclos, como foi apresentado no capítulo anterior. Em resumo, o método da divisão de dados serve principalmente para situações em que um *CR_link* é afetado, sendo possível tentar minimizar o impacto da falha por utilizar o primeiramente o remapeamento das tarefas. Quando interconexões entre

roteadores (*RR_links*) apresentam falhas, tenta-se primeiro utilizar o roetamento adaptativo, uma vez que é uma técnica pouco onerosa, mas não exclui-se a possibilidade de aplicar o remapemanto também em primeiro lugar, tentando mapear a interconexão defeituosa para uma região não (ou pouco) utilizada na rede. Quando o roteamento adaptativo não pode solucionar completamente o problema das falhas na rede, a divisão de dados pode e/ou deve ser também utilizada.

5.2 Remapeamento de Redes com Núcleos Heterogêneos

Quando os núcleos de uma rede intra-chip não são homogêneos, significa que eles não podem facilmente assumir tarefas uns dos outros. Para que seja possível a utilização de técnicas de remapeamento, é necessário utilizar técnicas que possibilitem a redundância de componentes da rede (núcleos, roteadores e interconexões em geral) para que regiões defeituosas possam ser substituídas, a fim de maximizar alguns compromissos como tempo de comunicação e latência, por exemplo. Normalmente, permitir o uso do remapeamento em redes heterogêneas implica em ter uma significativa quantidade de área extra a fim de permitir a redundância. Se por algum motivo o problema com área e eficiência energética não for uma restrição ao projeto tão significativa quanto o tempo de comunicação na presença de falhas, então a redundância pode ser aplicada com sucesso a fim de permitir o remapeamento das regiões utilizadas pela rede para prover a comunicação entre os núcleos.

Para aplicar o remapeamento em núcleos heterogêneos, colunas e linhas adicionais de roteadores e núcleos precisam ser inseridas na rede, proporcionando uma replicação de até três vezes o tamanho da rede original, para manter a mesma idéia de remapeamento utilizada com as redes homogêneas (permitindo a configuração da rede original, espelhada vertical, horizontal, e com o espelhamento de ambas, sem alterar a localização dos componentes). A replicação dos componentes pode ser realizada de acordo com a prioridade dos componentes que mais vulneráveis em caso de falha. A figura 7.7 ilustra alguns exemplos de redundância, e vamos considerar uma rede com 12 núcleos heterogêneos organizados originalmente como consta no canto superior esquerdo da figura 7.7. Se apenas a linha de núcleos do meio for considerada crítica em caso de falhas para a aplicação, ela pode ser facilmente replicada, aumentando a área da rede em torno de 1/3, já que mais uma linha de componentes será inserida. Então, seguindo essa idéia, podemos acrescentar linhas e colunas de componentes de acordo com a importância dos mesmos, no caso de existirem falhas na rede. O caso considerado máximo neste exemplo é quando a rede é complemente replicada por inserir mais 3 linhas e 4 colunas de componentes (último caso da figura 7.7). Nesta situação, a penalidade máxima acrescentada em área e energia é de 3 vezes a penalidade original da rede, porque o hardware extra é a triplicação do hardware original. A rede também pode ser interpretada como 4 redes distintas que estão acopladas, e toda a replicação utilizada

permite escolher a região com o mínimo número de falhas a fim de minimizar o impacto no tempo de comunicação. Para as falhas que não puderem ser evitadas, as técnicas de roteamento adaptativo e divisão de dados deverão ser utilizadas. As configurações escolhidas para o operabilidade da rede estão destacadas para cada situação, na figura 7.7.

Figura 7.7: Exemplos de replicação que permitem o remapeamento utilizando a topologia torus.

No exemplo apresentado na figura 7.7, o que se pode perceber é que no caso da rede onde foram utilizados 16 núcleos, os núcleos mais críticos que foram considerados para a duplicação foram: N5, N6, N7 e N8. Já no caso da rede com 20 núcleos, além dos 4 núcleos citados anteriormente, também foram duplicados os núcleos N3 e N11, e o núcleo N7 foi agora replicado 3 vezes. Desta forma, a duplicação dos componentes pode ser feita de acordo com a disponibilidade e a necessidade. Núcleos que possuem taxas de comunicação baixas não necessitam ser replicados tanto quanto núcleos que possuem altas taxas de comunicação, uma vez que o impacto na comunicação deles deve ser mínimo para a rede.

Quando se compara o impacto geral desta solução de mapeamento para núcleos heterogêneos com outros trabalhos abordados na literatura, é possível notar que não seria um absurdo triplicar a rede para obter bons compromissos entre alguns parâmetros. Os trabalhos de (OST et al., 2011) e também de (MANDELI et al., 2011), propõem soluções similares para utilizar o remapemamento dinâmico de tarefas a fim de lidar com o comportamento dinâmico da comunicação em redes intra-chip, com um

significativo impacto em área extra de quase 3 vezes o tamanho original da rede sem redundância.

Uma pergunta que pode surgir quando a rede é completamente triplicada é: qual a vantagem de projetar 4 redes em uma única rede em vez de projetar 4 redes que poderiam ser utilizadas separadamente, a fim de escolher dentre elas a rede que apresenta menos defeitos? A resposta pode ser considerada simples: porque de acordo com a localização das falhas, é possível obter "redes secundárias" às redes originais. Basta notar que existem linhas em destaque que separam cada quadrante, e que cada quadrante significa uma réplica da rede, no último caso da figura 7.7. No exemplo apresentado, existe pelo menos uma falha em cada quadrante, mas graças a disposição dos componentes foi possível encontrar uma rede secundária formada entre os 2 quadrantes da esquerda, que encontra-se entre a terceira e quinta linha de elementos. Com essa solução, é possível utilizar somente regiões livres de falhas da rede, e nenhuma técnica de tolerância a falhas permanente precisa ser mantida em funcionamento, uma vez que a região escolhida é livre de falhas.

Mesmo com posições variadas, é possível obter o mesmo mapeamento dos núcleos utilizando-se alguns componentes duplicados. Para procurar por redes secundárias dentro da rede projetada, basta verificar se as ligações entre cada núcleo conferem com as ligações originais. Por exemplo, basta avaliar se o núcleo N6 está conectado aos núcleos N2, N5, N7 e N10. Para todas as combinações apresentadas, a topologia torus foi considerada, e quando alguns componentes não são necessários eles podem ser desligados e contornados pela solução proposta no trabalho de STENSGAARD e SPARSO (2008), ou podem ser completamente desligados desde que não existam falhas na rede toleradas pelo roteamento adaptativo, uma vez que as conexões da malha de realimentação (torus) não poderiam ser utilizadas já que não estariam disponíveis com o desligamento de alguns componentes.

Existem ainda outras formas de solucionar o problema do remapeamento de núcleos heterogêneos. A forma mais simples e factível é colocar cada núcleo executando como um software descrito dentro de processadores, e utilizar apenas processadores dentro da rede intra-chip. Nas situações em que alguma memória é requerida, ela pode ser posicionada em lugares estratégicos que permitam a utilização delas com os mapeamentos disponíveis. Outra solução bastante simples é colocar mais de um núcleo por roteador, e utilizar multiplexadores para escolher qual núcleo irá utilizar a rede. Para esta última solução, o impacto em área na rede intra-chip é mínimo, considerando que apenas alguns multiplexadores serão necessários, embora exista a replicação de alguns núcleos. Assim, baseando-se na idéia de remapeamento para núcleos homogêneos, em uma rede com 12 núcleos, haveria a duplicação de 4 núcleos na linha de roteadores central, e os outros 8 roteadores (das extremidades superior e inferior da rede) teriam 4 núcleos cada um.

8. RESULTADOS DE SÍNTESE

Uma rede intra-chip SoCIN com topologia torus foi utilizada com buffers no canal de entrada do roteador (com capacidade para armazenar 4 flits). Para a primeira abordagem a tecnologia de 90 nm foi utilizada, enquanto que para a segunda abordagem, a tecnologia de 65 nm foi considerada.

8.1 Primeira Abordagem

Para os resultados de síntese, uma biblioteca *standard cell* de 90 nm com tecnologia CMOS foi utilizada com a ferramenta *Power Compiler* da *Synopsys*, na primeira abordagem deste trabalho, em que todos os fios podem ser defasados em até 50% em cada interconexão. Na tabela 8.1 os resultados de área extra, frequência máxima de operação e potência para 500 MHz e 8 bits de largura de canal são apresentados para as seguintes propostas:

Original→ Arquitetura de um roteador apresentada em (ZEFERINO e SUSIN, 2003), sem qualquer tolerância a falhas.
RRADD (sem DD ativo)→ Nossa proposta de arquitetura para um roteador funcionando apenas com o roteamento adaptativo. O bloco de DD está presente, mas foi desligado com a técnica de *sleep transistor* (LONG e HE, 2004) (SHI e HOWARD, 2006).
RRADD (com DD ativo)→ Nossa proposta de arquitetura para um roteador com ambas as soluções ativas para tolerar falhas: roteamento adaptativo combinado com divisão de dados.
Hamming→ Solução que utiliza código de Hamming para proteger cada interconexão de até uma única falha.

A interconexão utilizada em todas as propostas apresentadas é de 8 bits, exceto que para o caso do HC existem 4 bits extras em cada interconexão, a fim de enviar o código necessário para providenciar a detecção e correção dos dados em caso de falha. É muito importante lembrar que a proposta de RRADD permite lidar com múltiplas

falhas *interlink* e *intralink*, enquanto que Hamming pode lidar com apenas uma única falha em cada interconexão. Para redes com uma largura maior de dados, como 16 ou 32 bits, o impacto da nossa proposta original de divisão de dados torna-se tão inviável quanto o impacto de HC. Para interconexões de X bits (onde X é um número maior que 9), RRADD precisa de multiplexadores com até X entradas, aumentando consideravelmente o caminho crítico, e consequentemente limitando a frequência em um valor bem mais baixo que 500 MHz. Porém, tão desfavorável quanto nossa proposta para canais maiores do que 16 bits, HC precisa utilizar uma cascata de portas lógicas XOR, tendo uma cascata de pelo menos X/2 portas encadeadas na codificação e na decodificação dos dados, tornando a sua utilização também um forte sinônimo de hardware extra.

A área extra do RRADD é em torno de 28%, enquanto que HC tem 15% de área extra quando ambos são comparados com a versão original do roteador RASoC, respectivamente. Mas é muito importante observar que o número de interconexões extras não é levado em conta neste cálculo de área, apenas a área para a lógica é considerada. Para uma rede 4x3 em duas dimensões com a topologia torus, existem 72 interconexões (34 interconexões entre roteadores, 14 interconexões torus e 24 interconexões entre núcleos e roteadores). Uma rede com 8 bits de dados para comunicação e 4 bits de controle requer então 864 fios. Contudo, uma mesma rede que utilize roteadores com código de Hamming para a proteção contra falhas, necessita de 1152 fios, o que corresponde a 33,3% de fios extras nas interconexões para canais de 8 bits. Para interconexões com 16 e 32 bits, Hamming tem ainda 25% e 16% de fios extras, sem considerar ainda qualquer área lógica extra, mostrando que nossa proposta embora tenha mais área lógica consegue manter o mesmo número de interconexões.

RRADD apresenta também na tabela 8.1 uma frequência de operação maior do que a proposta que utiliza Hamming. Isso se deve ao fato de que Hamming utilizada uma cascata de portas XOR para codificar e decodificar o dado na saída e na entrada de cada canal do roteador. Enquanto isso, RRADD utiliza apenas multiplexadores na entrada e saída de cada canal, e quando DD não está ativo (porque não existem falhas que necessitam de DD para solucioná-las), apenas um multiplexador 2:1 é utilizado para contornar o bloco DD que é mantido desligado.

Com base nos resultados da tabela 8.1, RRADD sem DD ativo tem a mesma potência que o roteador original para uma frequência igual a 500 MHz. Quando o bloco DD está ativo no roteador, ele incrementa em torno de 45% os resultados de potência, enquanto que o roteador com Hamming incrementa 50% quando comparado com o roteador original. Isso significa que a atividade das portas XOR no caso do roteador com Hamming apresenta um consumo maior de potência do que os multiplexadores utilizados por RRADD para configurar os fios no bloco DD quando existem falhas na interconexão. Nossa proposta possui um consumo variável porque o bloco DD permanece ativo apenas quando existe alguma interconexão com fio(s) defeituoso(s) ao redor do roteador. Desta forma, para falhas entre roteadores que possam ser solucionadas pelo roteamento adaptativo, o bloco DD pode ser desligado sem qualquer prejuízo para a rede, enquanto que para os casos que RA não pode lidar, o bloco DD

deve permanecer ativo nas extremidades da interconexão defeituosa. Assim, para uma rede intra-chip com 12 núcleos (4x3) e dois roteadores com o bloco DD ativo, por exemplo, nossa proposta consome 18,34 mW, enquanto que a proposta com Hamming consome 25,44 mW, justamente por ter o código de Hamming ativo em todos os roteadores. Isso significa que Hamming incrementa o consumo total de potência da rede toda em 80% quando comparado a proposta original sem qualquer tolerância a falhas, enquanto que nossa proposta tem apenas 20% de excesso em potência em toda a rede, de acordo com os dados da tabela 8.1 calculados para uma rede com 12 núcleos. Isso significa que, dos 12 roteadores considerados na rede com a nossa proposta, 10 deles são RRADD sem DD ativo, e apenas dois com DD ativo.

Tabela 8.1: Comparações entre os resultados de síntese.

Roteador	**Area lógica (μm^2)**	**Frequência Máxima (MHz)**	**Potência @ Freq. Max. (mW)**	**Potência @ 500MHz (mW)**
Original	10.954	885	1,68	1,42
RRADD sem DD ativo	14.104 (+28%)	870 (-2%)	1,70	1,43
RRADD com DD ativo	14.104 (+28%)	588 (-33%)	2,41	2,07 (+45%)
Hamming	12.614 (+15%)	510 (-43%)	2,16	2,12 (+50%)

O impacto da potência nos fios de uma rede intra-chip também foi estimado, com simulações desenvolvidas através da ferramenta HSPICE, utilizando o modelo π distribuído (SAKURAI, 1983). Na tabela 8.2 são apresentados os resultados para uma variação nos dados de 500 MHz, para comprimentos de fios de 0,5 mm e 1 mm, sem o uso de repetidores. Para comprimentos de fio de 1 mm, pode-se perceber que a potência passa a apresentar valores significativos em relação a um roteador, sendo de 6 a 9 vezes a potência de um roteador sem TF para uma rede 4x3, e de 9 a 12 vezes para uma rede 4x4, para RRADD e Hamming, respectivamente. É importante ressaltar que os valores de potência no fio são os mesmos para o roteador original e para o RRADD, pois nenhum fio extra foi inserido na implementação.

Tabela 8.2: Resultados estimados de potência para todos os fios das redes 4x3 e 4x4.

Variação dos dados	**500 MHz**	**Potência (mW@500MHz)**	
		0,5mm	**1mm**
Rede 4x3	Original e RRADD (864 fios)	1,20	9,04
	Hamming (1152 fios)	1,64	12,06
Rede 4x4	RRADD (1152 fios)	1,64	12,06
	Hamming (1596 fios)	2,27	16,70

O que se pode notar na tabela 8.2 é que Hamming tem um impacto de 33% a mais na potência para o caso analisado, enquanto que nossa solução permanece com o mesmo impacto da proposta original, sem considerar a potência para a área lógica de qualquer técnica de TF.

8.2 Segunda Abordagem

Para a segunda abordagem, as combinações de entrada dos multiplexadores (que fazem parte do bloco DD) foram reduzidas em no máximo multiplexadores 8:1 na extremidade de cada canal. Nesta abordagem, o uso dos *latches* também foi incluído na transmissão dos dados, e os resultados foram avaliados para 8 e 32 bits de largura de canal, provendo então uma tolerância de 50 e 18,75% de fios defeituosos em cada interconexão, respectivamente. Uma biblioteca *standard cell* de 65 nm com tecnologia CMOS foi utilizada com a ferramenta *Power Compiler* da *Synopsys* para os resultados obtidos.

Para cada interconexão da rede, HC precisa de 4 e 6 fios extras por interconexão para enviar os valores da codificação, respectivamente, enquanto nossa proposta não tem esta necessidade, mostrando um considerável impacto no cálculo da potência que considera apenas as interconexões. Os resultados de síntese para cada estratégia são apresentados na tabela 8.3.

RRADD tem a maior quantidade de área extra por causa de suas propriedades de configurabilidade (*latches*, multiplexadores e controle dos mesmos). Mas a estratégia que utiliza HC tem maior impacto na frequência máxima, uma vez que os blocos codificadores e decodificadores possuem longas cadeias de portas XOR, considerando de 6 até 20 entradas para largura de canal entre 8 e 32 bits, respectivamente. Além disso, HC também apresenta um grande número extra de fios para enviar a codificação através das interconexões, influenciando posteriormente os resultados de energia. Resultados de potência na freqüência máxima e para 300 MHZ também são apresentados na tabela 8.3. Para RRADD existem dois possíveis resultados a serem considerados: quando o bloco DD está ativo no roteador, ou quando o bloco DD está desligado, o que significa que não existem falhas nas interconexões utilizadas pelo roteador, podendo então minimizar o impacto desativando a solução e melhorando os resultados de potência e energia, deixando o funcionamento do roteador limitado apenas ao roteamento adaptativo. Consequentemente, o roteamento adaptativo ainda pode lidar com algumas falhas nas interconexões com um mínimo de impacto na rede, o que faz dele a primeira técnica para ser utilizada. Apenas quando ele não pode lidar com alguns casos de falhas, é que a divisão de dados é empregada.

A funcionalidade dos *latches* foi validada pela prova de que o dado pode ser transmitido duas vezes mais rápido que a frequência máxima de cada roteador. Simulações a nível Spice foram realizadas para interconexões de 1 mm e 1,5 mm, valores considerados para o comprimento da interconexão de 8 bits e 32 bits,

respectivamente. Para a obtenção dos resultados, o modelo π distribuído foi utilizado para a representação do fio (SAKURAI, 1983). Para 1 mm, o atraso do fio corresponde a 0,22 ns, e para 1.5 mm o atraso corresponde a 0,35 ns. Com base nestes valores, podemos utilizar pelo menos duas vezes a interconexão dentro de apenas 1 período de relógio. Como pode ser observado na tabela 8.3, a frequência máxima que o roteador original pode atingir é limitada por 1.11 ns de período, o que significa que os valores de atraso do fio permitem a transmissão de dados pelo menos duas vezes dentro do ciclo de relógio, desde que o controle dos multiplexadores varie de acordo com o nível alto ou baixo do relógio.

Tabela 8.3: Resultados de síntese para a nova abordagem do roteador RRADD.

	Roteador	**Área (μm^2)**	**Atraso do Caminho Crítico (ns)**	**Max. Freq. (MHz)**	**Potência do Roteador (µW)@ Max. Freq.**	**Potência do Roteador (µW)@ 300MHz**	**# Total de Fios**
8 Bits	Original	5360	1.11	900	334.06	111.3	864
	AR	5260	1.11	900	338.19	112.7	864
	RRADD (DD on)	6978	1.71	585	498.49	255.9	864
	RRADD (DD off)	6978	1.71	585	216.64	112.7	864
	HC	5948	2.04	490	295.21	180.6	1152
32 Bits	Original	13850	1.26	793	811.04	306.8	2592
	AR	14071	1.26	793	819.12	308.4	2592
	RRADD (DD on)	19910	2.09	478	1392.5	873.6	2592
	RRADD (DD off)	19910	2.09	478	488.95	308.4	2592
	HC	16571	3.05	328	845.70	772.8	3024

Para o cálculo da energia que será apresentado no capítulo seguinte, a potência para cada fio foi calculada de acordo com a variação dos dados na rede. Dois benchmarks serão considerados para a análise de potência e energia: MPEG4 e VOPD. A seguir, na tabela 8.4 são apresentados os resultados de potência em cada fio da rede, para cada estratégia utilizada, e para cada benchmark considerado. Para cada situação, de acordo com a taxa de injeção dos flits na rede, existe uma taxa de variação diferente para os dados. Um pacote injetado corresponde a 4 flits na rede. Como os resultados foram considerados para um mesmo tempo de execução na rede, a taxa de injeção dos pacotes será a mesma para cada proposta, porque todas as propostas estão executando em 300 MHz. Por este motivo a potência no fio foi considerada a mesma para cada

estratégia, uma vez que a variação dos dados deve acontecer de acordo com a injeção dos pacotes, e esta por sua vez depende exclusivamente dos benchmarks utilizados.

Tabela 8.4: Resultados de potência em cada fio de acordo com cada benchmark utilizado.

Parâmetros	**MPEG4**		**VOPD**	
	Injeção de Flits @ 300 MHz	**Potência @ Injeção de Flits**	**Injeção de Flits @ 300 MHz**	**Potência @ Injeção de Flits**
8 Bits – 1 mm	3,3 ns	7.52 µW	4 ns	6.52 µW
32 Bits – 1,5 mm	13,2 ns	6.79 µW	16 ns	5.43 µW

9. RESULTADOS DE DESEMPENHO, ENERGIA E POTÊNCIA

Para as simulações deste trabalho, foram escolhidos benchmarks amplamente utilizados na literatura para compressão de áudio e vídeo, com padrões e taxas de comunicação bem diferenciados e diversificados, variando entre grandes congestionamentos e ausência de comunicação em diferentes regiões (BERTOZZI et al., 2005) (VU-DUC NGO et al., 2005). Cada um dos benchmarks é apresentado a seguir, nas figuras 9.1 para o VOPD, 9.2 para o MPEG4 e 9.3 para o H.264, e as simulações foram desenvolvidas ciclo-a-ciclo utilizando a implementação em VHDL baseada na rede SoCIN com uso da ferramenta ModelSim XE III 6.3.

Também foram analisados alguns padrões de tráfego sintético (DUATO, J. et al., 1997), como o padrão borboleta e o padrão complementar apresentados na figura 9.4, utilizando uma rede 4x4 para facilitar a visualização da comunicação, uma vez que a comunicação dos núcleos está relacionada com a posição do núcleo e com os seus respectivos pares ordenados. O padrão complementar é caracterizado por comunicar núcleos com as coordenadas binárias inversas, ou seja, cada um dos bits é substituído pelo bit oposto. Por exemplo, o núcleo "0000" comunica-se com o núcleo "1111", bem como o núcleo "1000" comunica-se com o núcleo "0111".

Já o padrão de tráfego borboleta caracteriza-se pela troca dos bits mais significativos pelos bits menos significativos. Dois exemplos de comunicação do padrão borboleta são os núcleos "1010" e "1011" que comunicam-se respectivamente com os núcleos "0101" e "1101". Como tanto o padrão borboleta quanto o padrão complementar utilizam coordenadas binárias, optou-se por utilizar uma rede de 16 núcleos a fim de permitir a correta comunicação dos núcleos com os respectivos tráfegos, evitando que alguns núcleos ficassem sem os seus respectivos pares.

Para cada caso simulado, foi necessário definir o intervalo de envio dos pacotes. Para os benchmarks do MPEG4, VOPD e H.264, utilizou-se a seguinte equação para definir o intervalo de envio dos pacotes:

*taxa_injecao_pacotes = frequencia_maxima(MHz)*largura_canal (Bytes)/taxa(MB/s)*

A equação apresentada calcula de quanto em quanto tempo (em ciclos) é necessário enviar um pacote para que a taxa de cada comunicação seja mantida. Para os padrões complementar e borboleta considerou-se a taxa de envio mínima para todas as comunicações (isto é, 1 pacote é enviado a cada 4 ciclos, já que o pacote é composto por 4 flits).

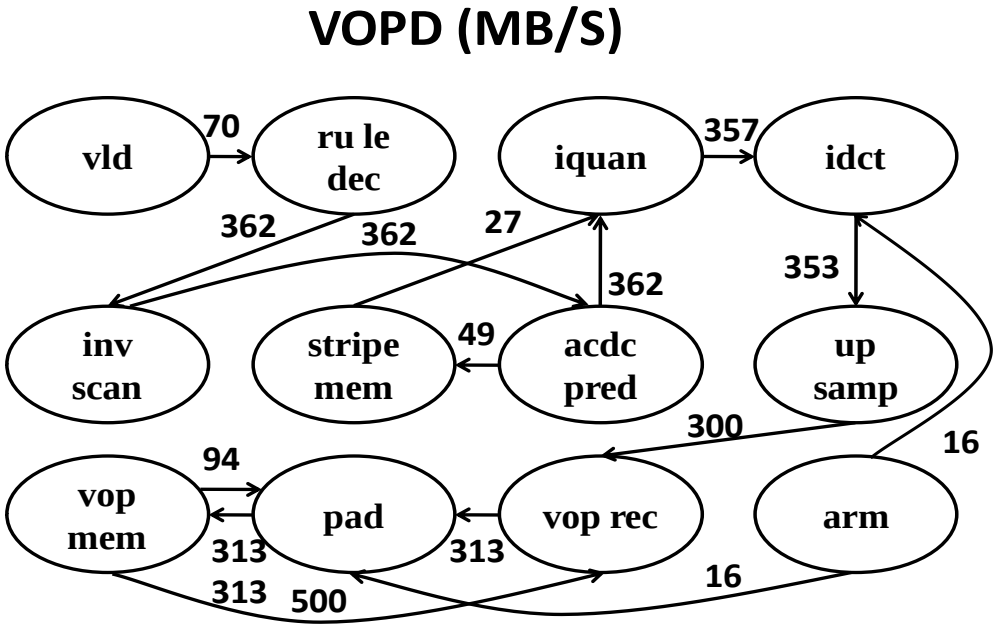


Figura 9.1: Padrão de comunicação do VOPD.

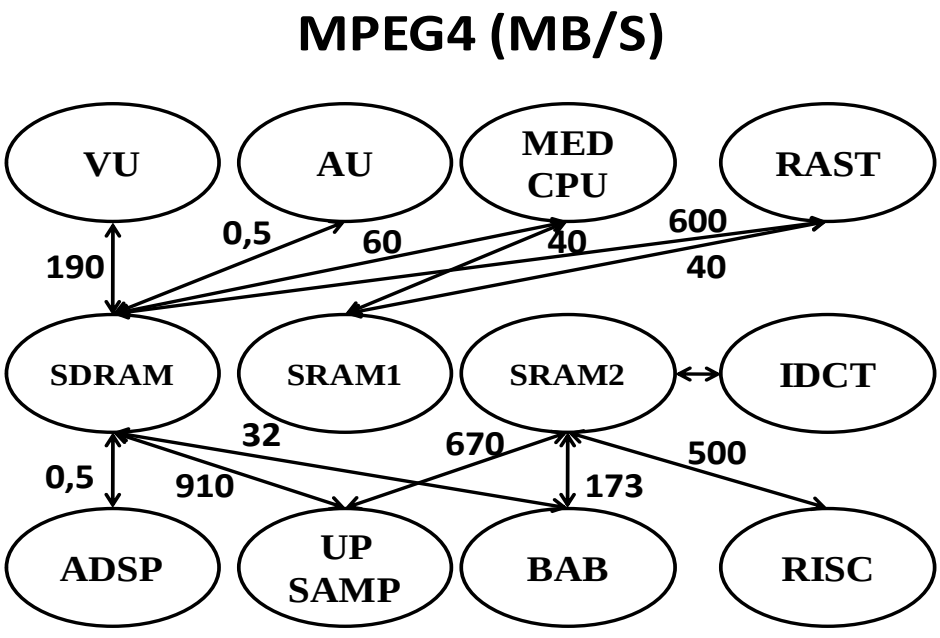


Figura 9.2: Padrão de comunicação do MPEG4.

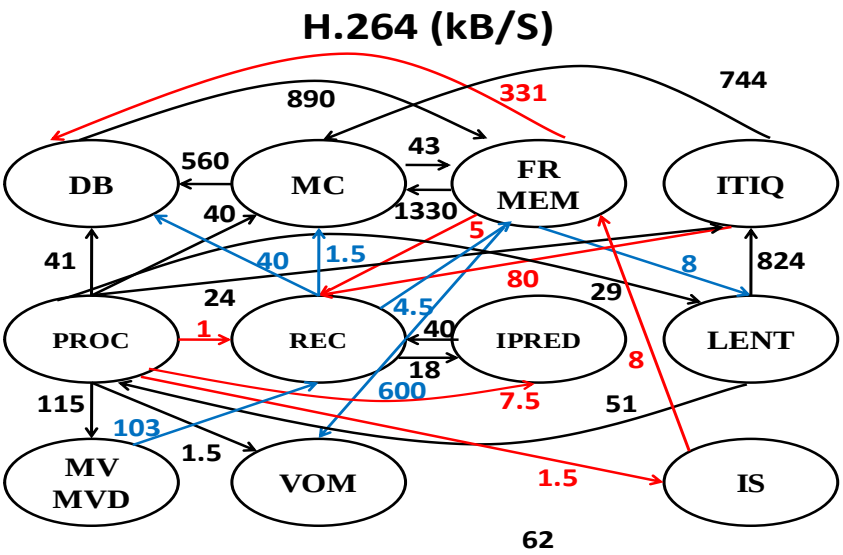


Figura 9.3: Padrão de comunicação do H.264.

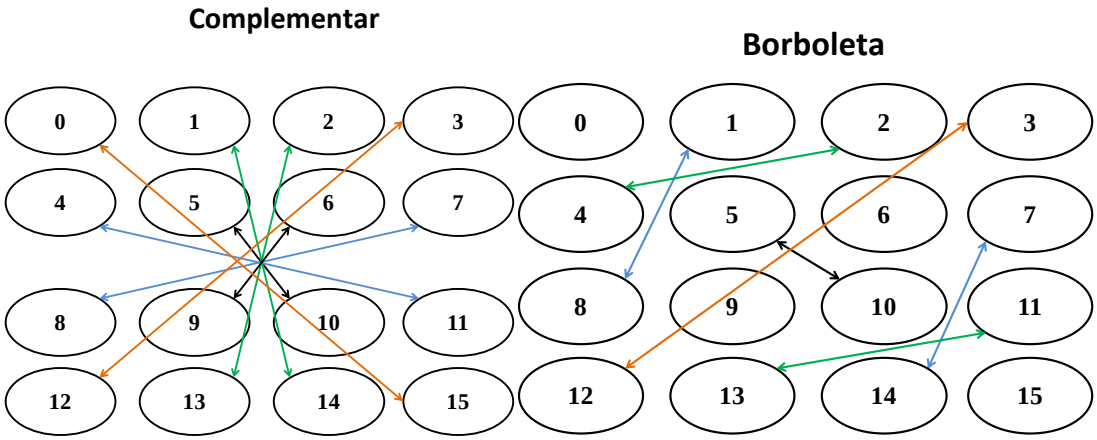


Figura 9.4: Padrões de tráfego sintético: complementar e borboleta.

Para analisarmos as falhas simuladas em cada interconexão, é necessário primeiramente apresentar a figura 9.5 em que cada interconexão é identificada para ser possível referenciá-la nos resultados obtidos. Na figura 9.5, as interconexões são divididas em 3 grupos: interconexões que ligam um rotador a um outro roteador, que ligam roteadores nas extremidades da rede (torus) e que ligam roteadores com núcleos, em uma rede de tamanho 4x3 com 12 núcleos. Para os resultados de simulação, foram consideradas apenas situações de falha única que podem ser resolvidas pelo roteamento adaptativo ou pela divisão de dados, considerando apenas falhas *intralink* (como explicado na seção 3.1). A opção de utilizar apenas falha única foi escolhida devido ao grande número de simulações necessárias para obter resultados de múltiplas falhas (*interlink*), já que é necessária a combinação de uma interconexão com todas as outras para poder analisar o comportamento de apenas uma situação.

Para uma rede 4x4 são ainda adicionadas as interconexões entre roteadores na vertical i35, i36, i37, i38, i39, i40, i41 e i42, e na horizontal i43, i44, i45, i46, i47 e i48, bem como as interconexões ITE4 e ITW4 para a malha de realimentação torus, e as interconexões in25, in26, in27, in28, in29, in30, in31 e in32 para conexão entre núcleos e roteadores. Em resumo, uma linha extra de roteadores é adicionada.

Para cada benchmark analisado numa rede 4x3 e 4x4, o número de interconexões utilizadas é apresentado na tabela 9.1. Nela, pode-se comparar o número total de conexões que a rede possui, e quantas conexões que a rede realmente está utilizando para prover a comunicação em cada benchmark.

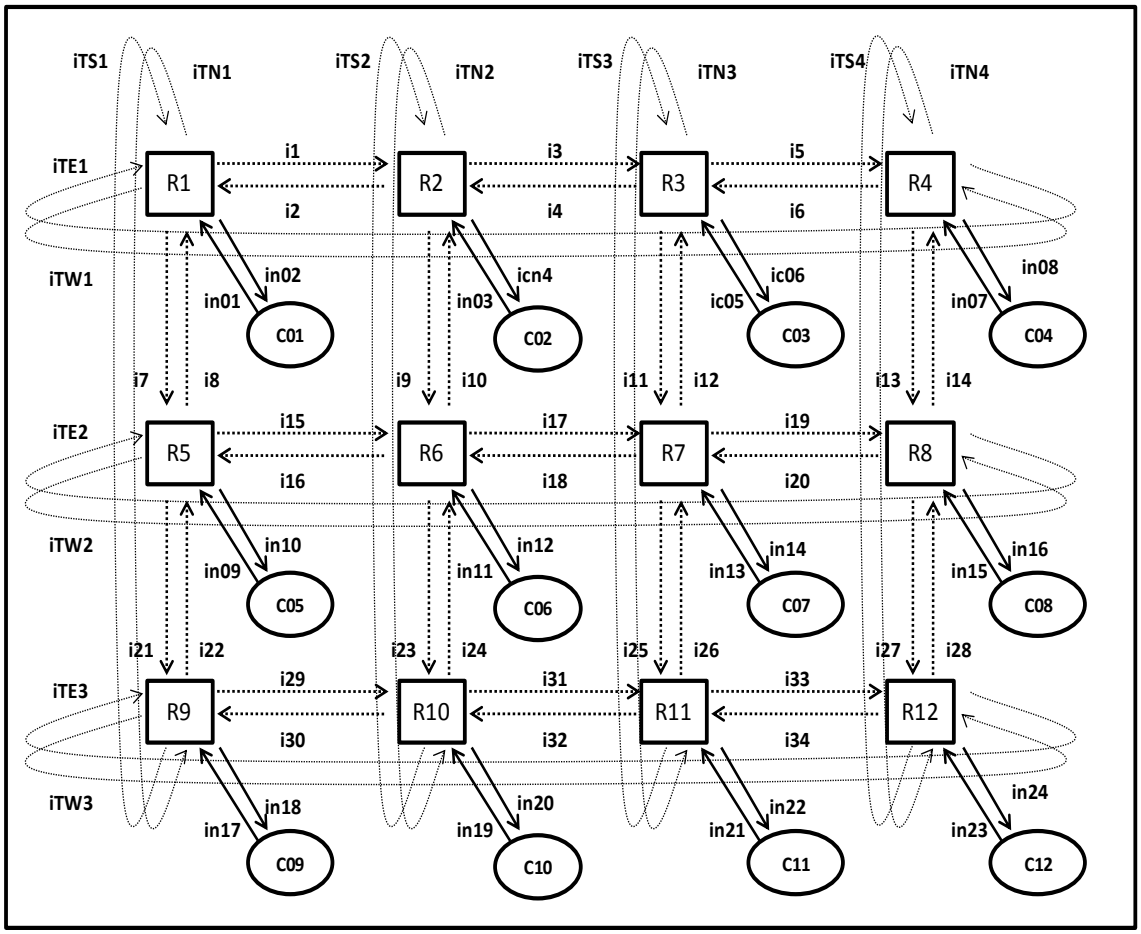


Figura 9.5: Nomenclatura das interconexões em uma rede 4x3.

O que se pode perceber na tabela 9.5 é que para o VOPD 54% das interconexões são utilizadas, enquanto que para MPEG4 e o H.264 a comunicação é provida por 70% e 61% das interconexões, respectivamente. Sendo assim, pode-se concluir que existem situações em que nem sempre uma falha na interconexão irá influenciar na comunicação, como é o caso de quando as falhas atingem a parcela de interconexões não utilizadas, que corresponde a 46%, 30%, e 39%, para o MPEG4, o VOPD e o H.264, respectivamente. Para os padrões de tráfego borboleta e complementar, apenas 42% e 60% das interconexões foram utilizadas, respectivamente.

Tabela 9.1: Número de interconexões utilizadas em uma rede 4x3 e 4x4.

	Interconexões Roteador-Roteador	**Interconexões Torus**	**Interconexões Roteador-Núcleo**	**Total**
Rede 4x3 Torus (total)	34	14	24	72
Rede 4x4 Torus (total)	48	16	32	96
Conexões usadas para VOPD	16	1	22	39 (54%)
Conexões usadas para MPEG4	24	2	24	50 (70%)
Conexões usadas para H.264	20	3	21	44 (61%)
Conexões usadas para o padrão borboleta	20	4	24	40 (42%)
Conexões usadas para o padrão complementar	14	12	32	58 (60%)

Para todos os casos analisados, foram considerados pacotes de dados com 4 flits, e a largura de canal utilizada é de 1 byte. Todos os pacotes enviados contém no primeiro flit o cabeçalho do pacote, com o endereço do destino, e os 3 flits seguintes contém a informação útil ao núcleo, incluindo a flag de flit finalizador no quarto e último flit de cada pacote.

Para todas as simulações, as comparações foram realizadas quando todos os núcleos efetuam suas comunicações enviando 1000 pacotes de informação para os seus respectivos destinos. Para o MPEG4 e o H.264 existem exceções: para o primeiro benchmark, nos casos em que as taxas são inferiores a 1MB/s, optou-se por enviar apenas 1 pacote a cada mil ciclos, e para o segundo benchmark, nos casos em que as taxas são inferiores a 10 kB/s enviam-se apenas 10 pacotes de dados a cada 1000 ciclos. Ambas as situações descritas foram adotadas de modo que comunicações mais lentas não prejudicassem completamente a execução da aplicação, sem retardar completamente o tempo de execução.

A frequência de operação utilizada para os benchmarks MPEG4 e VOPD foi a frequência máxima obtida pela rede, de 588 MHz quando a divisão de dados é utilizada, e de 880 MHz quando apenas o roteamento adaptativo é necessário, enquanto que para o Hamming a máxima frequência permitida é de 510 MHz em qualquer circunstância. Para o H.264 optou-se por utilizar 300 MHz, uma vez que não existem grandes taxas de comunicação, possibilitando uma redução na potência, sem qualquer prejuízo na execução, uma vez que as taxas são baixas o suficiente para serem executadas em 300 MHz.

Para todas as simulações foram considerados núcleos homogêneos. Isso é possível porque consideramos para cada núcleo um aplicativo (programa) correspondente que simula as tarefas físicas de um núcleo, sendo executado dentro de processadores que compõem a rede. Para cada um dos benchmarks utilizados, foi possível obter comparações e análises das interconexões que poderiam utilizar remapeamento, roteamento adaptativo, divisão de dados ou a combinação deles a fim de obter-se o mínimo impacto na comunicação. Falhas que aparecem nos *RC_links* sempre exigem a utilização de DD, e as vezes podem ser combinadas com o remapeamento para minimizar o atraso na comunicação. Falhas que ocorrem nos *RR_links* podem ser na maioria das vezes solucionadas pelo simples uso do roteamento adaptativo, e quando esta opção não é possível, a divisão de dados pode ser empregada. Se o roteamento adaptativo ou a divisão de dados em *RR_links* apresentar um grande impacto na comunicação, a comunicação através de interconexões defeituosas também pode ser solucionada com o uso do remapeamento. Neste trabalho, para simplificar a combinação das técnicas utilizadas, primeiro aplicou-se o roteamento adaptativo nas falhas localizadas em *RR_links* e analisou-se o impacto. Após o tratamento das falhas nos *RR_links*, analisou-se a solução de remapeamento para falhas em *RC_links*, onde a solução de DD precisa ser utilizada para garantir a funcionalidade do núcleo mapeado para o local defeituoso.

9.1 Situações Toleradas pelo Roteamento Adaptativo na Primeira Abordagem

Primeiramente, se analisou casos em que apenas o roteamento adaptativo é suficiente para resolver o problema de uma interconexão defeituosa. Consequentemente, nem todo hardware projetado foi necessário (bloco DD), uma vez que o roteamento adaptativo é suficiente para solucionar os casos de falhas entre os roteadores abordados nesta seção. Como resultado disso, a solução de DD pode ser desativada no circuito, permitindo que a rede seja executada em 880 MHz, porque o caminho crítico foi minimizado.

Para o VOPD, existem apenas 17 interconexões utilizadas para prover a comunicação, dentre 48 interconexões que poderiam utilizar o roteamento adaptativo (34 interconexões roteador-roteador e 14 interconexões torus - que ligam as extremidades). Das 17 interconexões utilizadas, apenas 12 delas afetam a comunicação de algum núcleo da rede, como é exibido no gráfico da figura 9.6.

O número de núcleos afetados por alguma falha entre roteador-roteador para o VOPD pode variar em até 4 núcleos no pior caso de comunicação, isto é, quando uma falha existe significa que até 4 núcleos podem sofrer algum impacto no tempo de comunicação. Quando o roteamento adaptativo é utilizado, o impacto no tempo de execução da aplicação (VOPD), considerando as comunicações afetadas, pode ser de

0,1 até 2,2% para o caso simulado. Quando essas porcentagens são colocadas em termos de valores de acordo com o número de ciclos de execução necessário para o roteamento adaptativo, a comparação que pode ser feita é que nossa solução irá ter um tempo extra de até 135 us no pior dos casos (para 118305 ciclos) para situações que utilizem roteamento adaptativo, enquanto que Hamming tem um tempo de comunicação constante de 232 us (para 116007 ciclos) e na maioria das vezes não pode corrigir falhas do tipo *intralink*, já que estas falhas correspondem a combinação errônea de dois fios dentro de uma mesma interconexão (Hamming só pode tolerar um fio falhado em cada interconexão), e nem reduzir a quantidade de hardware utilizada a fim de diminuir a potência e aumentar a frequência de operação.

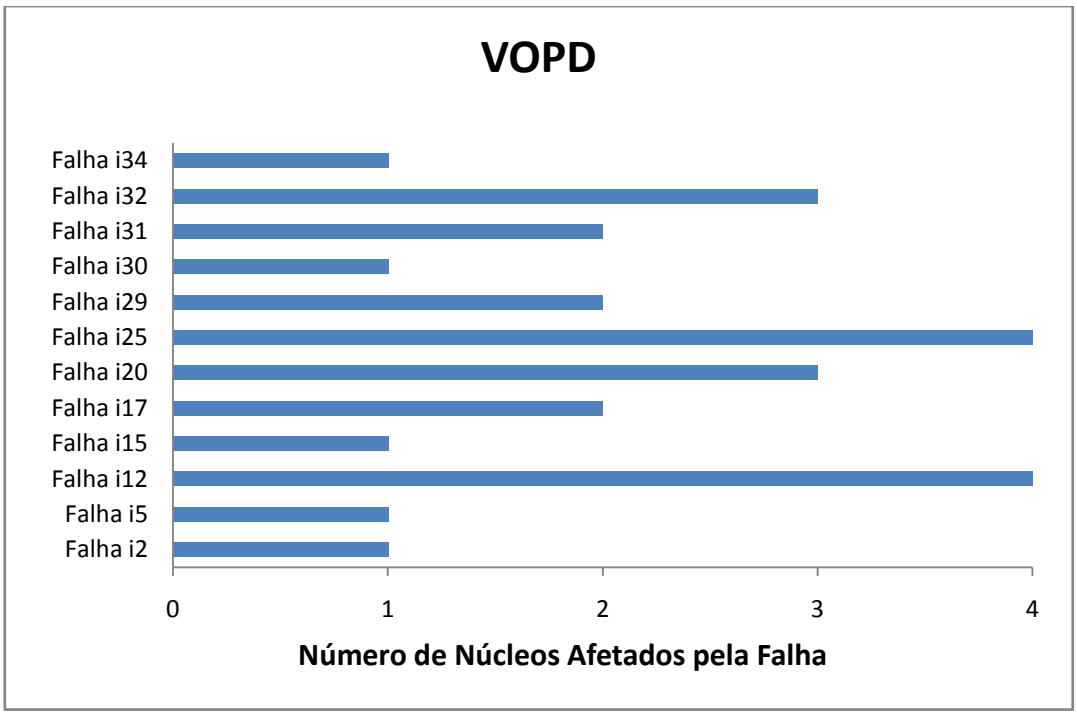


Figura 9.6: Quantidade de conexões afetadas pelo roteamento adaptativo para o VOPD.

Para o MPEG4, como mostrado na tabela 9.1, existem 26 comunicações que podem ser afetadas por uma falha, e que podem ser reparadas através do roteamento adaptativo. Dentre estas 26 interconexões, 50% delas não afetam o tempo de comunicação quando existe alguma falha localizada nelas, o que significa que mesmo sendo necessário um caminho alternativo, o número de ciclos para a comunicação não é afetado. As outras 13 interconexões podem afetar o tempo de comunicação, variando entre 0,2 e 6,6 % no pior dos casos. As 13 interconexões que podem causar algum impacto no tempo de comunicação por causa de falhas são exibidas no gráfico da figura 9.7, e no pior caso a comunicação pode afetar até 9 núcleos. Quando comparamos o pior tempo de comunicação com roteamento adaptativo de 83,4 us (73096 ciclos) com o tempo de execução de 130 us para o código de Hamming (66681 ciclos), pode-se observar que a proposta utilizada com o roteamento adaptivo sempre é mais atrativa do

que a opção de utilizar Hamming, uma vez que Hamming tem um impacto constante no tempo de comunicação.

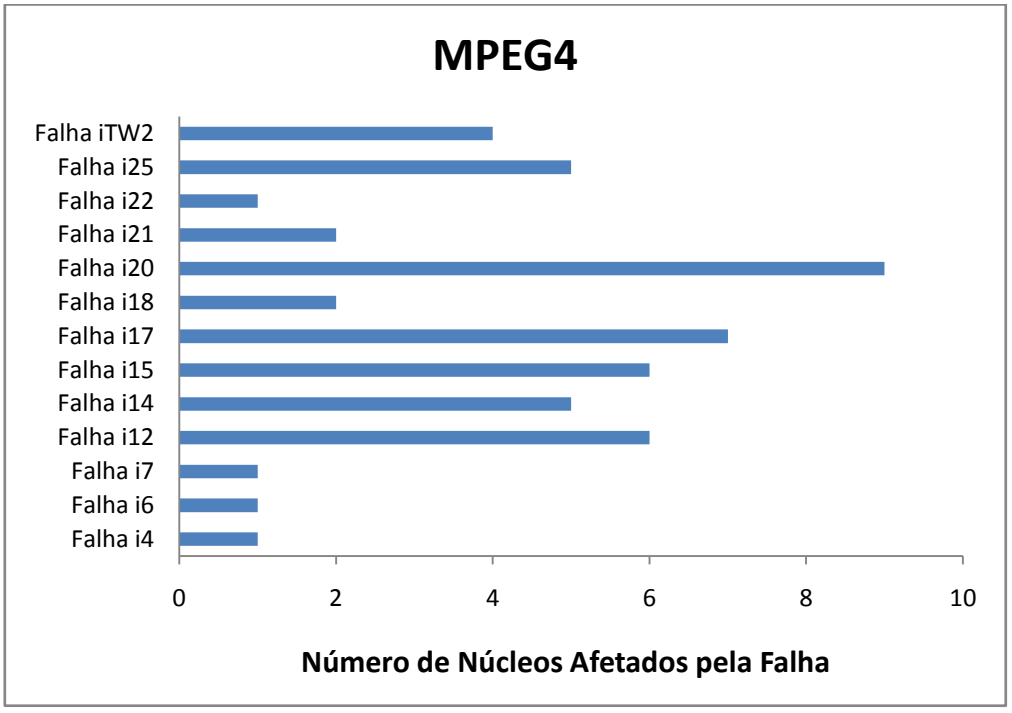


Figura 9.7: Quantidade de conexões afetadas pelo roteamento adaptativo para o MPEG4.

Como se pode observar na figura 9.3 (benchmark do H.264), o padrão de comunicação do H.264 possui núcleos com um elevado número de alvos para realizar a comunicação, enquanto que o número de comunicações utilizadas não é muito maior que os outros benchmarks (61%). Logo, pode-se concluir que existem mais comunicações compartilhando as mesmas interconexões. Consequentemente, o número de comunicações afetadas por uma falha qualquer entre os roteadores torna-se um pouco maior, ainda mais com a utilização de caminhos alternativos através do uso do roteamento adaptativo.

Das 23 interconexões utilizadas, 15 delas mostraram que afetam a comunicação de pelo menos 1 núcleo na presença de falha, como mostra a figura 9.8. Em relação ao tempo de comunicação, o impacto para o caso avaliado foi de até 3% a mais no tempo de comunicação, considerando que qualquer implementação foi executada em 300 MHz, uma vez que as taxas do H.264 são menores do que os outros benchmarks analisados. Sendo assim, neste caso não é possível obter ganhos do Hamming em tempo de execução, pois tanto o Hamming quanto a nossa proposta foram executados em 300 MHz. Assim, o roteamento mostrou ter um acréscimo de 3% no tempo de computação devido a necessidade de um caminho alternativo. Porém, é importante ressaltar novamente que o Hamming não pode cobrir todas as falhas toleradas pela nossa proposta, e que ambas as simulações foram executadas na mesma frequência devido às baixas taxas do H.264.

Para tornar a comparação entre os resultados dos benchmarks mais justa, os resultados são também apresentados lado a lado na tabela 9.2, para a primeira abordagem descrita neste trabalho. Todos os resultados são relacionados ao número de interconexões da rede que podem utilizar roteamento adaptativo, o que representa um total de 48 das 72 interconexões de uma rede 4x3. Para fins de comparação com outras soluções apresentadas na literatura, também foram colocadas comparações com Hamming, mesmo sabendo que devido a sua baixa eficiência para tolerar múltiplas falhas nem sempre é possível utilizá-lo.

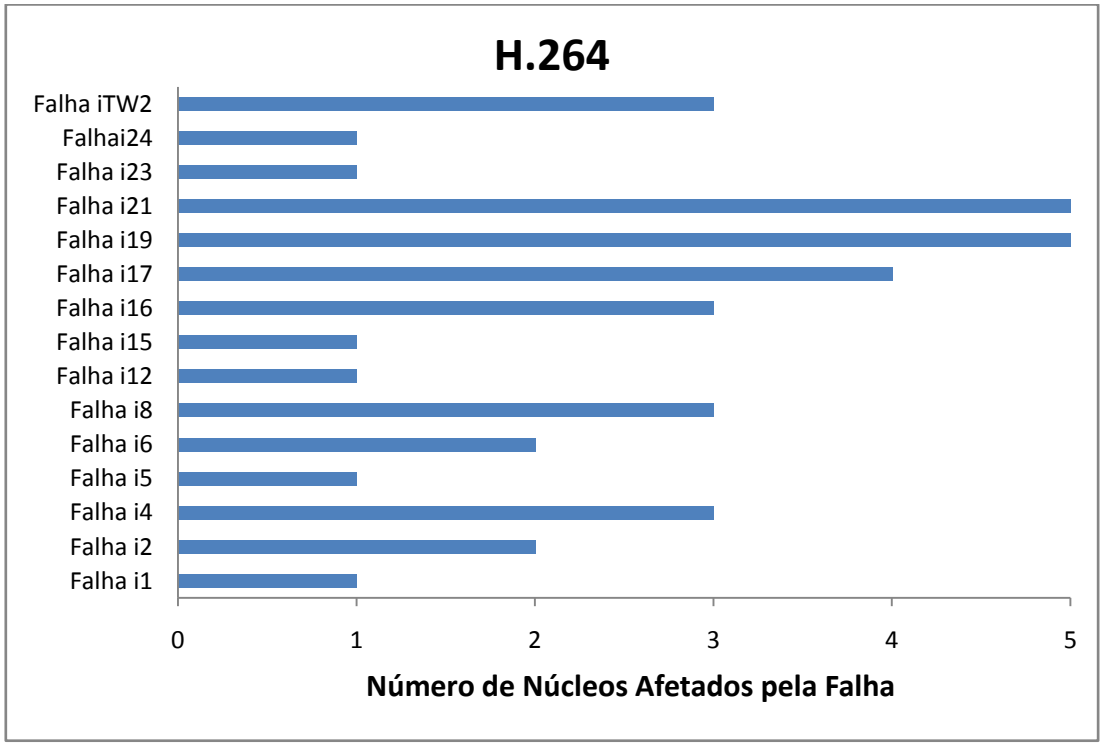


Figura 9.8: Quantidade de conexões afetadas pelo roteamento adaptativo para o H.264.

Tabela 9.2: Comparações entre alguns resultados de RA para os 3 benchmarks analisados.

	VOPD	**MPEG4**	**H.264**
Total de interconexões utilizadas	17 (35%)	26 (54%)	23 (47%)
Máximo de núcleos afetados pelo RA	4 (33%)	9 (75%)	5 (41%)
Máximo impacto apresentado no tempo de comunicação	2,2%	6,6%	3%
Frequência de operação	880 MHz	880 MHz	300 MHz
Frequência de operação com Hamming	510 MHz	510 MHz	300 MHz
Pior tempo de Computação	135 us	84,3 us	202 us
Tempo de computação com Hamming	232 us	130 us	196 us

Também foram realizadas simulações para uma rede 4x4 com padrão de tráfego sintético complementar e borboleta. Para o padrão complementar, quando a rede 4x4 é simulada sem qualquer falha, nenhuma das comunicações interfere nas comunicações

alheias. Ou seja, nenhum caminho da rede se cruza, quando não existem falhas. Das 64 interconexões que podem permitir a utilização do roteamento adaptativo, somente 26 são utilizadas. Destas 26 interconexões, 8 delas não apresentam qualquer impacto quando são detectadas como defeituosas, porque o roteamento adaptativo utiliza um caminho alternativo que não está sendo utilizado por nenhuma outra conexão. Para as outras 18 interconexões, a penalidade na comunicação será sempre a mesma: impacto na comunicação de 2 núcleos. Isso acontece porque o roteamento adaptativo precisa utilizar interconexões que já estão sendo utilizadas por alguma outra comunicação na rede, e acaba tendo que compartilhá-las. Assim, o caminho alternativo acaba dobrando o tempo de comunicação, pois onde passavam 1000 pacotes, com o uso do RA será necessário passar o dobro de 1000, pois existem 1000 pacotes que estão desviando a interconexão defeituosa. Assim, o impacto no tempo de comunicação é afetado em 100% para todas as 18 interconexões consideradas (de 9 us para 18 us), enquanto que com Hamming o impacto foi de 16 us.

Para o padrão borboleta, os resultados mostraram que 20 interconexões entre os rotadores e 4 interconexões torus são utilizadas na rede quando a rede executa sem qualquer tipo de falha. Para a simulação de uma falha em cada uma dessas interconexões, os resultados mostraram que existem 4 situações que o roteamento adaptativo não afeta qualquer tempo de comunicação, 10 situações que apenas a comunicação de 1 núcleo é afetada, 7 situações que afetam pelo menos 2 núcleos e apenas 3 situações que afetam a comunicação de 3 núcleos. Se o tempo de comunicação total da aplicação for levado em conta, que é de 18,1 us, em 62% dos casos que o roteamento adaptativo foi utilizado o tempo de comunicação subiu para 26,7 us, mostrando uma penalidade de quase 50% no tempo de comunicação. Para as outras 9 interconexões (38% dos casos) o tempo de comunicação total não foi afetado (é importante ressaltar que uma comunicação afetada não significa necessariamente um impacto no tempo de comunicação). Quando comparamos com Hamming, o tempo de comunicação é sempre penalizado em 32 us.

A tabela 9.3 mostra um comparativo entre os dois tráfegos sintéticos analisados e a implementação que utiliza HC. Como se pode perceber, o padrão complementar é o único caso avaliado que pode apresentar uma pequena desvantagem de 12% da nossa abordagem em relação ao HC, quando compara-se o tempo de computação das aplicações. Porém, a abordagem proposta com RRADD nem sempre terá o pior tempo de computação na presença da falha. Como dito anteriormente, existem 8 situações para o padrão complementar que o uso de RA não afeta nenhuma comunicação, bem como 9 situações para o padrão borboleta que não afetam o tempo final da computação.

Tabela 9.3: Comparações entre alguns resultados de RA para os 2 tráfegos sintéticos analisados.

	Complementar	**Borboleta**
Frequência de operação com RRADD	880 MHz	880 MHz
Frequência de operação com Hamming	510 MHz	510 MHz
Total de interconexões utilizadas	26 (40%)	24 (37%)
Interconexões utilizadas não causam impacto no tempo	8	9
Máximo de núcleos afetados	2 (12%)	3 (18%)
Melhor tempo de computação	9 us	18,1 us
Pior tempo de computação	18 us	26,7 us
Tempo de computação com Hamming	16 us	32 us

9.2 Situações Toleradas pela Divisão de Dados na Primeira Abordagem

Para as simulações realizadas com divisão de dados considerando a primeira abordagem, as falhas foram consideradas apenas entre um roteador e um núcleo (ou vice-versa), e a frequência utilizada foi de 588 MHz para a nossa proposta, enquanto que para o código de Hamming a frequência foi mantida em 510 MHz.

Para o VOPD, os resultados mostram que o impacto da divisão é mais significativo para apenas 1 caso de falha simulada (na interconexão do núcleo ARM), uma vez que o ARM é o núcleo com a taxa de comunicação mais baixa. Mesmo quando nenhuma falha é simulada na rede, o núcleo ARM é ainda o núcleo com o maior impacto no tempo de comunicação, uma vez que possui a taxa de comunicação mais baixa da rede e todos os outros núcleos precisam esperá-lo terminar a comunicação. Para todos os outros casos de falha (nos outros núcleos), o impacto foi considerado baixo, afetando apenas a comunicação de 1 ou 2 núcleos. O impacto no tempo de comunicação foi de apenas 2,5%, porque existe um núcleo muito lento no VOPD que limita a comunicação (ARM). Desta forma, mesmo que exista uma falha em alguma interconexão considerada com altas taxas de comunicação, a falha em questão pode nem sequer afetar o tempo de comunicação, já que a comunicação rápida (com falha) pode acabar antes da comunicação lenta (sem falha).

Somente a fins de comparação, foram somados todos os tempos de comunicação de cada comunicação finalizada. Ou seja, para cada comunicação estabelecida, levou-se em conta o tempo de comunicação quando o último pacote foi transmitido e recebido com sucesso por cada par de fonte e destino, e ao final de todas transmissões, todos os tempos foram somados. Para todos os casos de falha simulados nas interconexões entre um núcleo e um roteador para o benchmark do VOPD, o tempo máximo de execução obtido, considerando a soma de todas as comunicações, foi de 757 us. Quando nenhuma falha atinge a rede, este tempo de execução é de 738,3 us, mostrando que o impacto

total na rede é então muito baixo, em torno de 2% quando analisado desta forma. Quando comparamos a mesma situação com Hamming, o tempo de execução sempre será de 868 us, já que a codificação está sempre presente na rede, mostrando que o impacto no tempo total de todas as comunicações pode ser de até 17%. Todos estes resultados estão ilustrados no gráfico da figura 9.9.

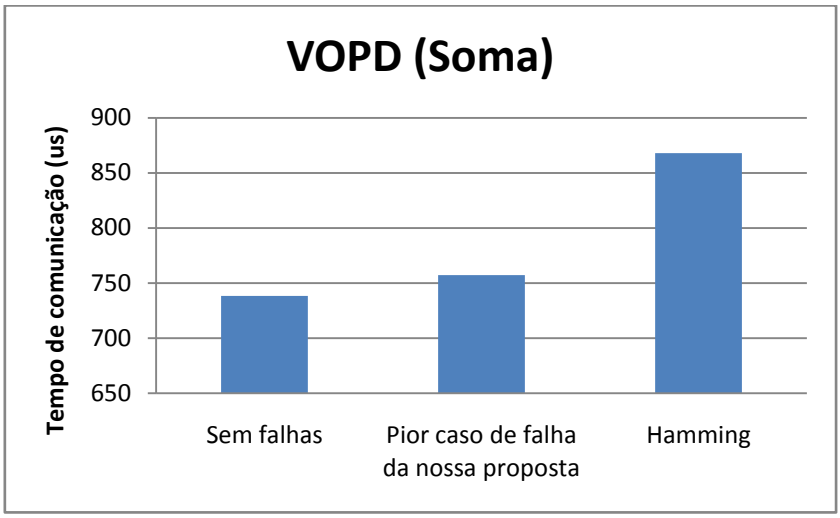


Figura 9.9: Impacto na soma dos tempos de comunicação para 3 situações.

Quando o MPEG4 é avaliado em relação ao impacto das falhas entre núcleos e roteadores, o que se pode notar é que no pior caso de falha até 8 núcleos podem ter suas comunicações prejudicadas. Mesmo quando o pior caso não é manifestado, ainda assim tem-se um alto impacto na comunicação, como é o caso das falhas no *RC_link* que afetam diretamente os núcleos 7, 8 e 10, por exemplo, como mostrado na figura 9.10. Os núcleos que foram omitidos não apresentam um impacto significativo.

Em relação ao tempo de computação do MPEG4 para o cenário proposto, o que se pode verificar na figura 9.11 é que o impacto é muito diversificado. Existem alguns casos em que uma falha entre núcleo e roteador nem sequer afeta o tempo de comunicação, como é o caso do núcleo 1, 2, 3, 4 e 9. Para os outros casos, o tempo de comunicação varia entre 113,4 us quando não existem falhas até 146,4 us no pior caso de falha (núcleo 5, que contém um elevado número de comunicações). Quando comparam-se os resultados com Hamming, o que se pode ver é que apenas o núcleo 5 acaba tendo um tempo de comunicação com maior impacto do que Hamming. Hamming, por sua vez, sempre tem um impacto constante de 133,7 us, enquanto que a solução proposta possui uma média de impacto de 118,5 us quando todas situações de falha entre núcleo e roteador são consideradas.

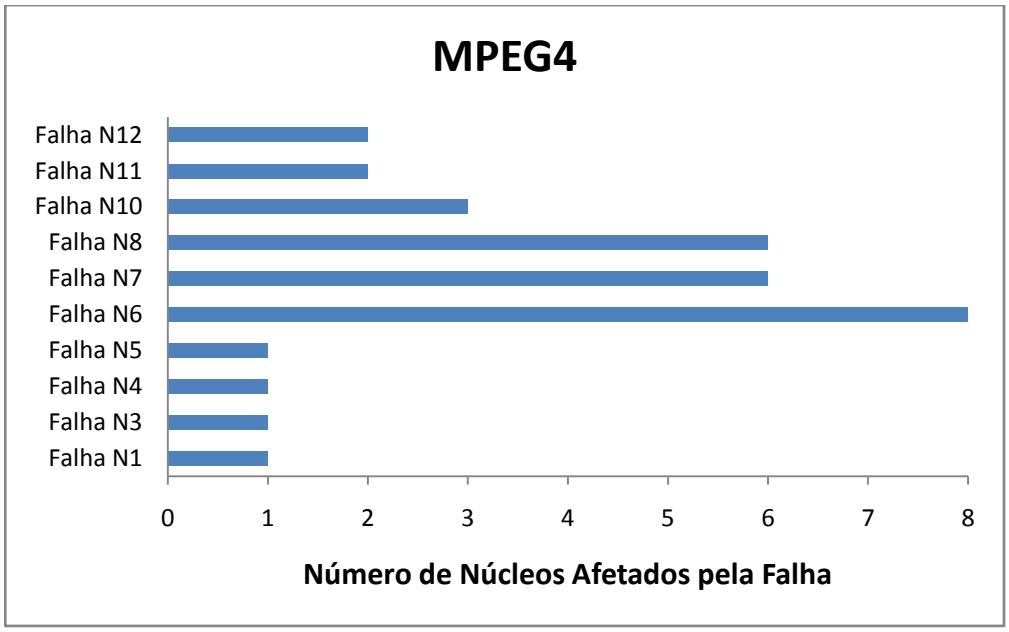


Figura 9.10: Número de núcleos que são afetados pela falha no *RC_link* entre o núcleo e seu respectivo roteador.

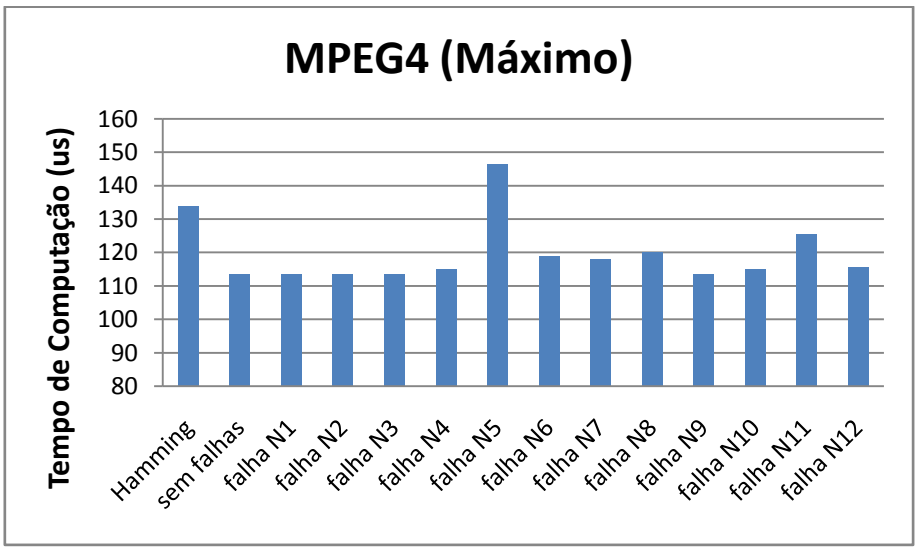


Figura 9.11: Impacto no tempo final de execução da aplicação de acordo com a localização da falha.

Quando se apresenta a soma dos tempos de cada comunicação para o benchmark do MPEG4, obtém-se o tempo de 821,6 us quando nenhuma falha é simulada na rede, como a figura 9.12 mostra. Para o pior caso de falha, a soma do tempo de todas as comunicações é de 1004 us, enquanto que a solução com Hamnming leva 889,3 us. A explicação para um impacto tão significativo em relação ao tempo sem falhas (de quase 20%) é porque, como foi apresentado na figura 8.9, existem até 8 núcleos afetados com o caso de falha entre o roteador e o núcleo 5. Sendo assim, cada comunicação realizada

através da interconexão defeituosa tem um impacto específico, mostrando que esse impacto pode ser grande quando o tempo de cada comunicação da rede é somado, chegando em torno de 20%. Quando comparado com Hamming, existem apenas 2 situações que apresentam um tempo extra significativo: núcleo 5 e núcleo 7. Contudo, quando consideramos a média, a nossa proposta ainda apresenta ganhos significativos: tempo de 855 us contra 889 us do Hamming.

Os resultados da divisão de dados para o H.264 são bem simples, uma vez que quase não afetam o tempo de comunicação. A variação no tempo de comunicação foi em torno de 1% quando se compara o tempo máximo de execução sem falhas (198 us), exceto pelo caso em que ocorrereu 15% de variação (229 us) para o núcleo REC, justamente por ser o núcleo com o maior número de comunicações. Essa variação também se assemelha para o caso da soma total de cada tempo de comunicação (sendo de 1472 us sem falhas), onde as variações são também de até 1%, exceto para o núcleo 5 que mostrou uma variação de 18% (1810 us). O número de núcleos afetados por alguma comunicação que passa pelo caminho defeituoso é apresentado na figura 9.13, e mostra uma variação de 1 até 5 núcleos afetados, mesmo que a variação tenha sido em geral de apenas 1%.

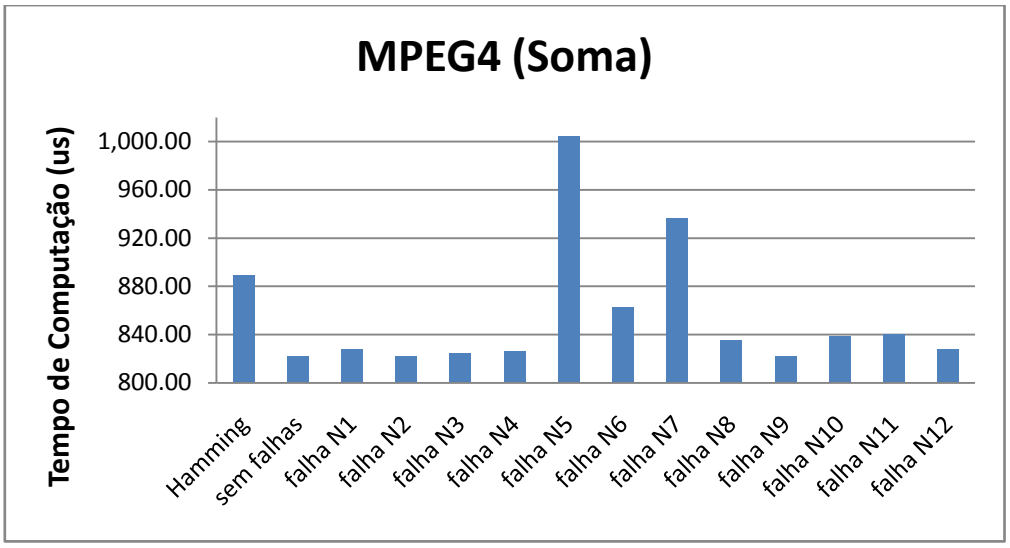


Figura 9.12: Impacto na soma de todos os tempos de cada comunicação de acordo com a localização da falha.

Os resultados de falhas para o padrão complementar que utiliza DD como solução é bastante peculiar: como cada uma das comunicações não compartilha ou afeta outra comunicação, o impacto acontece apenas na comunicação que tem a presença da falha. Sendo assim, cada interconexão entre roteador e núcleo que estiver defeituosa, somente irá afetar uma única comunicação, dobrando o tempo de comunicação da mesma, fazendo com que o tempo de execução da aplicação dependa exclusivamente do término da comunicação que utiliza a interconexão defeituosa. Para este caso, então o tempo de comunicação será de 27,2 us para qualquer conexão que apresente falha,

enquanto que para o Hamming o tempo de comunicação será sempre de 16 us, independente da ocorrência ou não de uma falha.

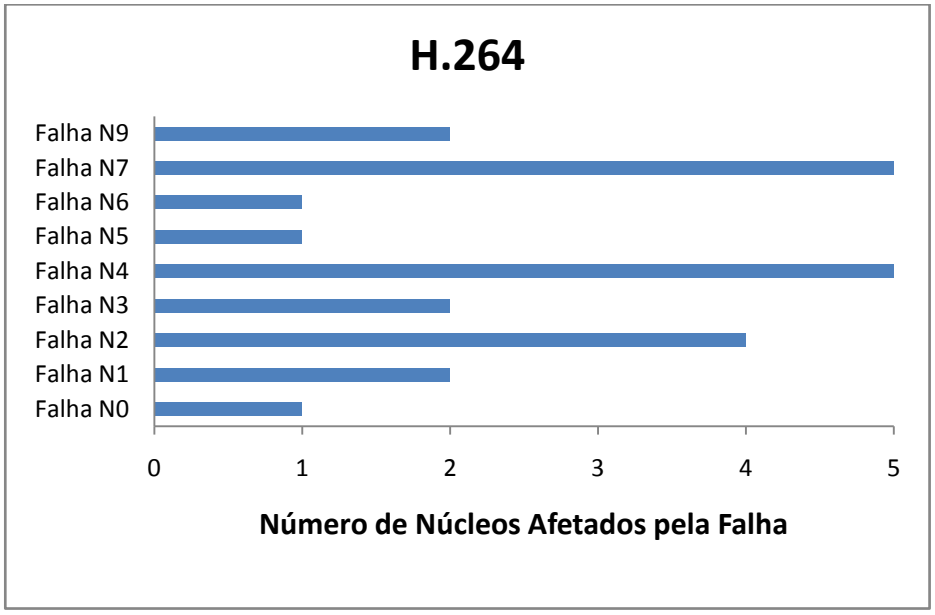


Figura 9.13: Número de núcleos que são afetados pela falha no *RC_link* entre o núcleo e o seu respectivo roteador.

No padrão borboleta, existem 4 núcleos que não efetuam a comunicação (N0, N6, N9 e N15), então apenas 12 núcleos são utilizados numa rede intra-chip com uma organização de 4x4 roteadores, sendo que apenas duas situações afetam exclusivamente a sua própria comunicação (N3 e N12, que possuem apenas 2 roteadores de distância entre o emissor e o receptor sem compartilhar qualquer comunicação), enquanto que as outras 12 situações afetam a comunicação de 2 núcleos, uma vez que os pacotes atrasam o envio daqueles que compartilham um mesmo caminho, mostrando um aumento no tempo de comunicação, de 27 us para 40 us. Para o caso do Hamming, o tempo máximo de computação ficou em 32 us, mostrando uma desvantagem na divisão de dados.

Quando o tempo de cada comunicação na rede é somando, como se pode observar na figura 9.14, é possível notar que o impacto do Hamming acontece em todas as situações, enquanto que nossa proposta permite otimizar as situações de falhas, onde apenas as extremidades da comunicação defeituosa é que são envolvidas no uso de DD, de forma a prejudicar o mínimo possível as comunicações da rede, considerando o padrão borboleta.

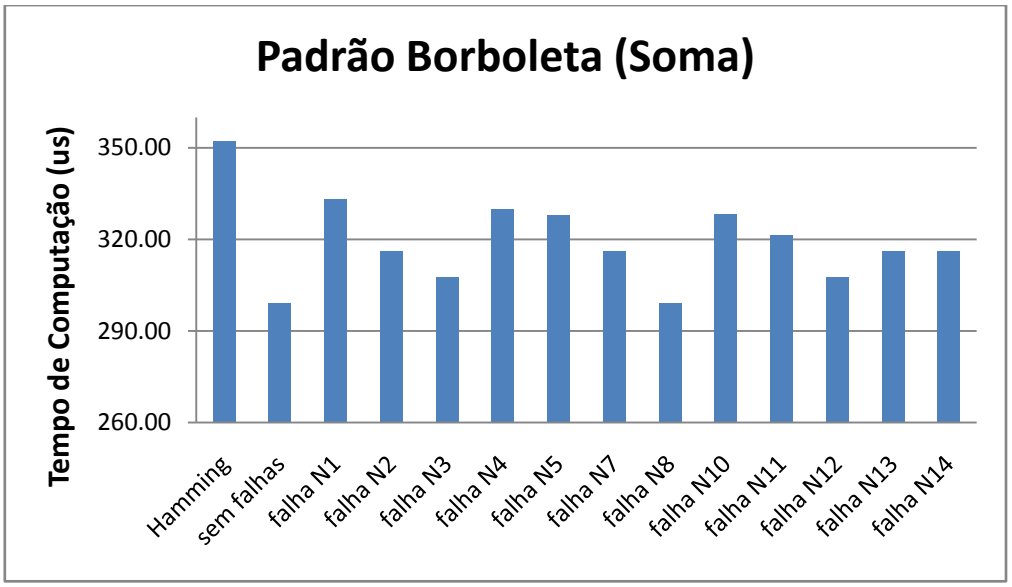


Figura 9.14: Impacto na soma dos tempos de cada comunicação de acordo com a localização da falha.

9.3 Energia e Potência para a Primeira Abordagem

Para tornar a comparação mais justa, optou-se por utilizar a rede com a frequência máxima de cada implementação (588 MHz para a solução proposta e 510 MHZ para o HC). Quando nenhuma solução de tolerância a falhas é utilizada, é possível que a rede execute em até 885 MHZ. Utilizando técnicas de "*Sleep Transistor*" (LONG e HE, 2004) (SHI e HOWARD, 2006) é possível desligar o bloco DD e multiplexadores são utilizados para contornar a região com TF. Assim as soluções de TF da nossa abordagem podem ser desligadas a fim de minimizar a energia dissipada. Para cada um dos casos a seguir o tempo médio de comunicação foi considerado.

Para os resultados de energia foram considerados os benchmarks do MPEG4 e do VOPD, e os tráfegos sintéticos complementar e borboleta. O benchmark do H.264 não foi considerado, uma vez que a frequência de operação foi considerada de 300 MHZ para ambas as propostas: nossa abordagem e Hamming. O tempo de comunicação considerado para o cálculo da energia foi estipulado através da média dos tempos de computação (em cada caso de falha, para cada tipo de falha), de acordo com a classificação que segue:

1) Caso I - corresponde a implementação original sem qualquer proteção contra falhas, apenas para comparar a penalidade que o acréscimo das técnicas de tolerância a falhas pode causar no tempo de execução das aplicações. O roteador dessa situação executa em 885 MHZ.

2) Caso II - nesta situação existe RA e DD, porém a técnica de DD não está ativa e um multiplexador é utilizado para contornar o bloco DD que foi desligado. Nenhuma

situação de falha foi simulada nesta configuração, que serve apenas para mostrar o impacto de contornar o bloco DD. A frequência de operação deste caso é de 870 MHz.

3) Caso III - neste caso, são consideradas apenas falhas que podem ser toleradas através do roteamento adaptativo. Por isso, o bloco DD continua desligado, e a frequência de operação é ainda de 870 MHz.

4) Caso IV - o roteador contém o bloco DD ativo em todos roteadores da rede, mas nenhuma falha é considerada. Essa implementação restringe a execução da rede em 588 MHz.

5) Caso V - são consideradas falhas que podem ser toleradas apenas com o bloco DD, porque o RA não pode lidar com elas. Como a utilização do bloco DD está sendo feita apenas por 2 roteadores, a dissipação de energia pode ser reduzida na rede, considerando que os outros blocos DD estão desligados.

6) Caso VI - nesta situação, foram consideradas as penalidades do caso III e do caso V para obter-se a média do tempo de comunicação. Para esta situação, existem 2 possibilidades que podem ser consideradas para caracterizar o tipo de falha que atinge a rede: a primeira delas, que considera 2 falhas *intralink* atingindo interconexões diferentes, ou então a segunda possibilidade que corresponde a uma falha *interlink* que é localizada na rede entre uma interconexão núcleo-roteador e uma interconexão roteador-roteador. Se ao invés de uma falha *interlink*, forem consideradas duas falhas *intralinks* entre as mesmas interconexões, Hamming já não tem o poder de corrigi-las, apenas detectá-las, uma vez que a falha *intralink* caracteriza 2 fios defeituosos em uma mesma interconexão. A frequência de operação é de 588 MHz.

7) Caso VI - roteador com código de Hamming. Este roteador tolera até 1 única falha em cada interconexão (apenas 1 fio defeituoso por interconexão), e tem sempre a frequência de 510 MHZ, uma vez que corrige as falhas em tempo de execução.

Para o caso I, II e IV não foram consideradas falhas na rede. Para os casos III e V apenas uma falha *intralink* foi considerada por simulação. No caso VI, as simulações não foram realizadas, apenas o tempo médio de computação foi calculado baseado na junção dos casos III e V. Para cada caso, o tempo de computação é apresentado na tabela 9.4.

Tabela 9.4: Tempo médio total de comunicação para cada caso analisado.

Caso de falhas	**Tempo de Comunicação @ Frequência Máxima (us)**			
	MPEG4	VOPD	Complementar	Borboleta
Caso I	75,4	131,0	9,0	18,0
Caso II	76,7	133,4	9,2	18,4
Caso III	77,0	133,7	18,4	24,4
Caso IV	113,3	197,2	13,6	27,2
Caso V	119,2	200,1	27,2	35,7
Caso VI	119,7	200,6	34,0	44,6
Caso VII (Hamming)	133,3	232,0	16,0	32,0

A tabela 9.5 apresenta resultados de potência para cada tipo de rede analisada, e os resultados finais de energia constam na tabela 9.6. Com base nos resultados apresentados, é possível concluir que a nossa solução apresenta sempre bons resultados de potência (ou equivalentes) com relação ao Hamming. Para as situações em que as taxas são bem variadas na rede (MPEG4 e VOPD), nossa solução apresentou bons resultados também em energia, mostrando uma melhora em torno de 25% em relação ao Hamming. Isso porque a potência é elevada apenas nos roteadores que utilizam o bloco DD, e naqueles roteadores onde não existem falhas ao redor, o impacto acrescentado na potência é mínimo, apenas porque alguns multiplexadores foram adicionados.

Tabela 9.5: Potência para cada rede analisada.

Caso de falhas	**Potência@Máxima Frequência (mW)**	
	4x3	4x4
Caso I	29,20	38,94
Caso II	29,44	39,26
Caso III	29,44	39,26
Caso IV	37,96	50,62
Caso V	30,86	40,68
Caso VI	30,86	40,68
Caso VII (Hamming)	37,98	51,26

Tabela 9.6: Resultados de energia, considerando o comprimento médio dos fios de 1 mm.

Caso de falhas	**Energia (uJ)**			
	MPEG4	VOPD	Complementar	Borboleta
Caso I	2,20	3,82	0,26	0,52
Caso II	2,25	3,92	0,27	0,54
Caso III	2,26	3,93	0,54	0,71
Caso IV	4,30	7,48	0,51	1,03
Caso V	3,67	6,17	0,84	1,10
Caso VI	3,69	6,19	1,05	1,37
Caso VII (Hamming)	5,06	8,81	0,60	1,21

Porém, para as redes que utilizaram taxas idênticas de comunicação (neste caso, os tráfegos sintéticos), os resultados da nossa solução não apresentaram vantagens na

implementação com relação ao tempo de comunicação, potência e energia. Isso aconteceu porque mesmo usando o RA ou DD, o impacto nas interconexões acaba sendo severamente prejudicado uma vez que existe o desvio de um caminho devido a falha ou o dobro do tempo de comunicação. O desvio afeta as comunicações que estão no caminho alternativo, pois agora com a falha existe mais uma comunicação para compartilhar o caminho. Como o intervalo de envio dos pacotes foi considerado o mínimo possível, não existiram ciclos em que a comunicação estava ociosa, e por isso o tempo de comunicação foi severamente prejudicado, sendo sobrecarregado para atender a mais uma comunicação em determinados caminhos utilizados como alternativa a falha.

9.4 Remapeamento para a Primeira Abordagem

Se considerarmos que os núcleos podem mudar de posição na rede sem qualquer degradação, e que existem pelo menos 4 configurações de mapeamento a serem escolhidas, trocando-se facilmente as posições de acordo com figura 7.5 apresentada na seção 7, então é possível que o tempo de computação seja reduzido, e como consequência a energia também será reduzida. Para tanto, vamos considerar a seguinte nomenclatura como ilustrado na figura 7.5:

- Mapeamento original;

- Espelhamento vertical (V);

- Espelhamento horizontal (H);

- Espelhamento vertical e horizontal (V & H).

Para todos os casos utilizados que foram apresentados na seção anterior sobre a energia, a configuração de mapeamento considerada foi a configuração original. Para cada tipo de falha analisada, pode existir uma configuração de mapeamento que minimize o impacto no tempo de comunicação. A seguir, algumas situações aleatórias foram escolhidas para mostrar o impacto do remapeamento, na tabela 9.7, em que TM significa tipo de mapeamento, TTC significa tempo total de comunicação e E significa energia. Quando os resultados são comparados com HC, a localização da falha não importa para os resultados de tempo e energia, uma vez que a técnica de HC é aplicada em todas as interconexões.

Para os exemplos apresentados, o remapeamento foi escolhido de forma que a presença das falhas foi mapeada para interconexões não utilizadas pela rede ou para interconexões com menor impacto na comunicação (quando não é possível remapear para uma interconexão não utilizada), a fim de melhorar o tempo de comunicação total da rede e consequentemente a energia. Para a rede complementar não foi possível obter melhorias, uma vez que todos os núcleos se comunicam com a mesma intensidade.

Tabela 9.7: Resultados do remapeamento para alguns casos analisados.

Benchmark / Padrão	**Antes do Remapeamento**			**Depois do Remapeamento**		
	Localização da Falha	**TTC (us)**	**E (uJ)**	**TM**	**TTC (us)**	**E (uJ)**
MPEG4	ITW2	80,6	2,37	H	76,6	2,25
MPEG4	I20	84,6	2,49	V	76,6	2,25
MPEG4	IN09	146,4	4,53	V	119,2	3,67
MPEG4	IN21	125,2	3,86	H	113,3	3,49
MPEG4	Hamming	133,3	5,06	-	133,3	5,06
VOPD	I32	135,7	3,99	H	133,4	3,92
VOPD	IN23	206,5	6,37	V / H /V & H	197,2	6,08
VOPD	Hamming	232,0	8,81	-	232,0	8,81
Borboleta	I7	27,0	0,79	H	18,4	0,54
Borboleta	IN25	40,8	1,26	H	27,2	0,89
Borboleta	Hamming	32,0	1,21	-	32,0	1,21

Tipo de Mapeamento (TM); Tempo Total de Comunicação (TTC); Energia (E)

Na figura 9.15 um comparativo é apresentado para a dissipação de energia de cada um dos casos exibidos na tabela 9.7, e o que se pode concluir é que nossa proposta apresentou uma boa eficiência energética em relação ao Hamming, que é sempre utilizado em tempo de execução para corrigir falhas permanentes e transientes. Nossa proposta mostrou que mesmo tendo 28% de área extra, é possível reduzir a energia da rede desligando partes que não precisam da tolerância a falhas, uma vez que as falhas já foram diagnosticadas na rede. De acordo com a figura 9.15 e a tabela 9.7, a energia após o remapeamento pode ser reduzida em quase 20% quando comparada com nossa proposta antes do remapeamento (interconexão IN21) e até 50% quando comparada com a energia da implementação que utiliza Hamming (interconexão I32). Isso demonstra que o remapeamento pode, sempre que possível, ser empregado para tentar minimizar os atrasos das comunicações com falhas.

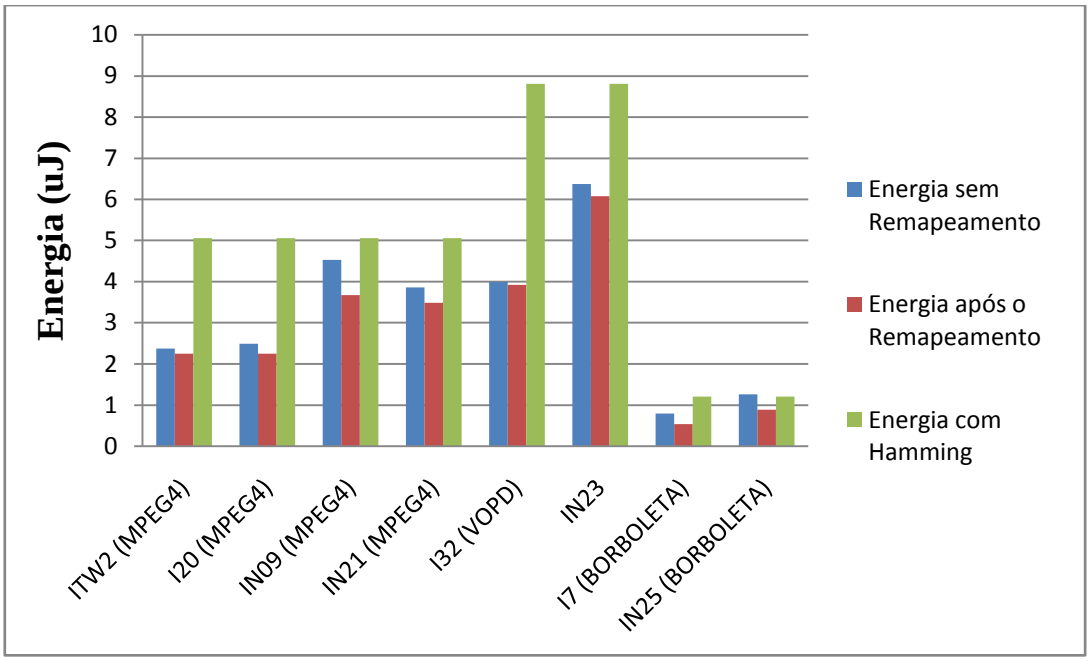


Figura 9.15: Comparativo de energia entre as propostas abordadas após a utilização do remapeamento.

9.5 Energia e Potência para a Segunda Abordagem

Para a estratégia que combina a utilização dos *latches* com o controle dos multiplexadores que constituem a divisão de dados, o tempo de comunicação acaba não sendo afetado pelo uso do bloco DD quando a frequência para todas as abordagens é normalizada para um mesmo valor: 300 MHz, por exemplo. Para um tempo de comunicação correspondente a 1000 segundos, todas as abordagens devem ter a mesma taxa de injeção de pacotes para cada benchmark simulado, de acordo com a largura de canal escolhida. Então na figura 9.16 a energia necessária para cada estratégia é apresentada para os benchmarks com 8 bits de largura de canal, enquanto que na figura 9.17 a energia é apresentada para 32 bits de largura de canal.

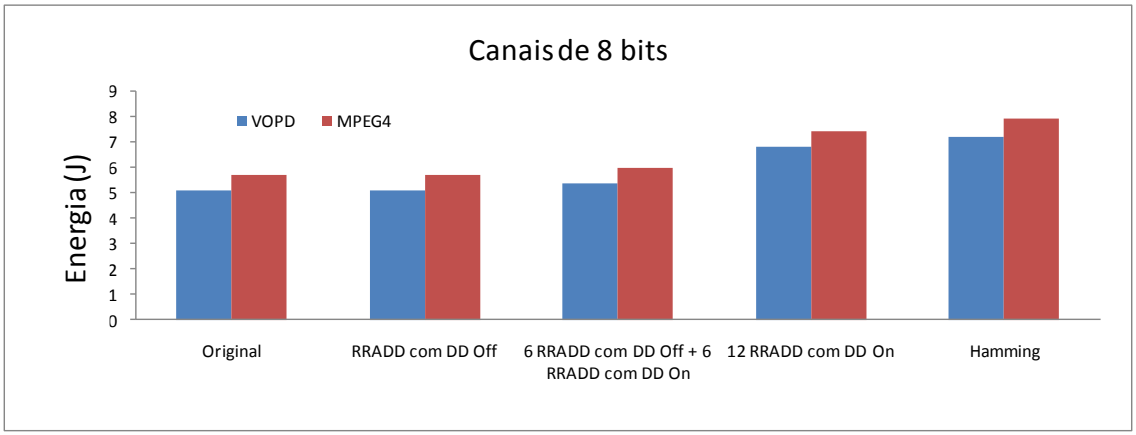


Figura 9.16: Comparativo de energia entre cada abordagem analisada para 8 bits.

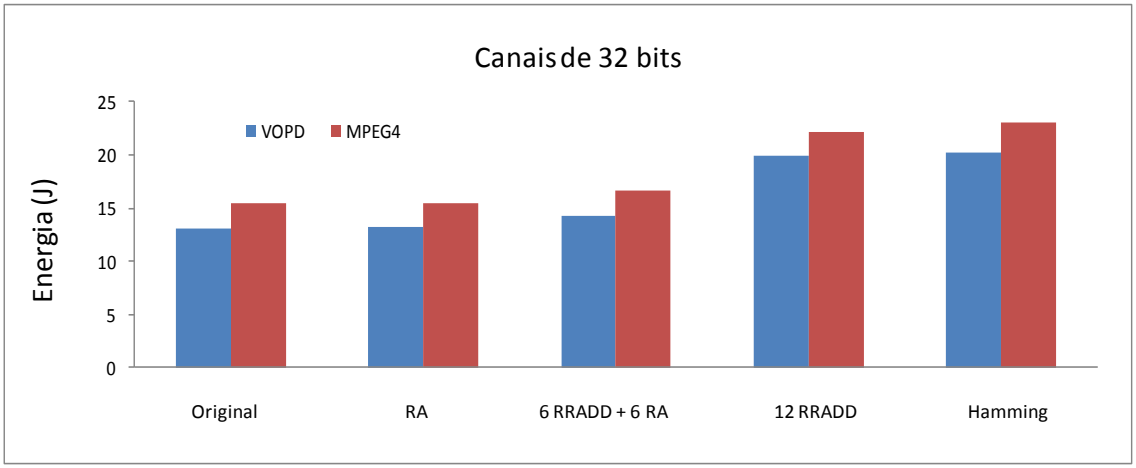


Figura 9.17: Comparativo de energia entre cada abordagem analisada para 32 bits.

Na figura 9.18, pode-se perceber que o impacto da proposta com RRADD é variável, de acordo com o número de interconexões defeituosas na rede, enquanto que o HC e o RRADD sem o bloco DD ativo possuem um valor constante de impacto na potência. Como conseqüência o mesmo impacto se reflete na energia, independente da quantidade de falhas registradas na rede. O que se pode concluir então é que o impacto da nossa proposta será sempre um valor variável ente 5% e 34 % no caso de canais com 8 bits, podendo variar entre 8% e 39% para 32 bits, enquanto que o impacto do Hamming sempre será de até 42% e 54%, respectivamente, para 8 e 32 bits. O pior caso para nossa proposta acontece quando todos os roteadores precisam utilizar o bloco DD, então o maior impacto em potência e energia será registrado.

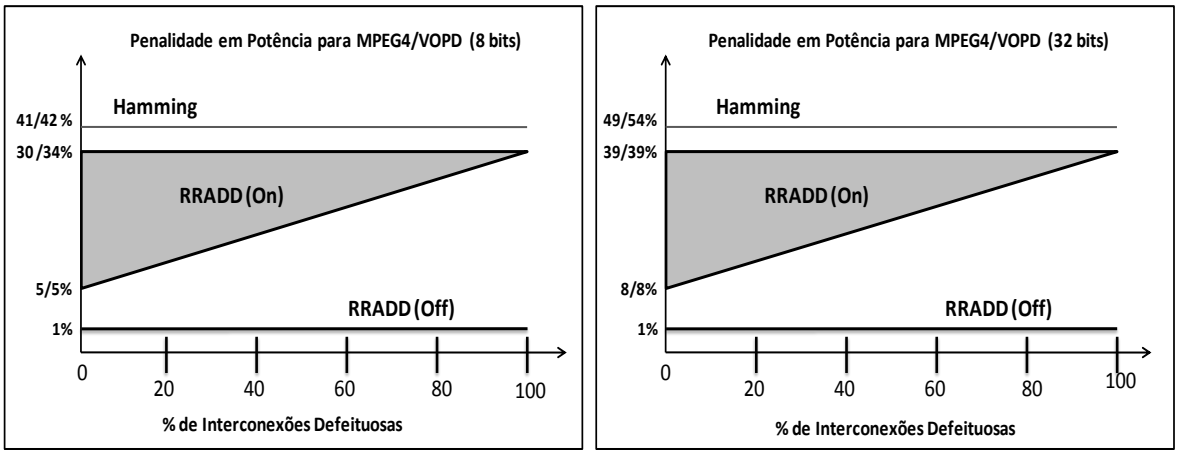


Figura 9.18: Impacto em % de energia para cada estratégia analisada.

10. ANÁLISE DE CONECTIVIDADE NA PRESENÇA DE MÚLTIPLAS FALHAS

No trabalho de (KAKOEE et al., 2011) foi utilizada a redundância em alguns componentes no roteador e nas interconexões para prover confiabilidade. Um BIST foi incluído na implementação para prover o diagnóstico da falhas. Os resultados de área mostraram um aumento de 12,5 até 15,5% de área, de acordo com o número de buffers utilizados. Porém, este trabalho apresenta um baixo percentual de conectividade quando o número de falhas incrementa na rede, chegando a apenas 30% de conectividade quando existem até 100 falhas numa rede 8x8, considerando que a área ocupada pelas interconexões é em torno de 5% da área da rede.

A figura 10.1 mostra a conectividade da rede de acordo com o número de falhas nas interconexões da rede intra-chip. O número de falhas nas interconexões varia de 0 até 100 falhas, distribuídas em qualquer fio da rede. Foram avaliados o melhor e o pior cenário de conectividade para um caso genérico, com o emprego de uma rede 4x3 e uma rede 8x8, ambas com a topologia torus e 8 bits de largura de dados.

O melhor cenário corresponde ao caso onde as falhas estão completamente distribuídas entre as interconexões, permitindo a comunicação com o mínimo de prejuízos para a rede. Se numa rede 4x3 existem 72 interconexões, o melhor cenário implica em poucas falhas em cada interconexão da rede, concentradas em interconexões onde o roteamento adaptativo pode tranquilamente ser utilizado, sem ultrapassar 50% de falhas em cada interconexão. Sendo assim, o melhor caso de falha pode corresponder a 4 fios defeituosos em uma interconexão, o que implica em 25 interconexões com falha, para um canal de 8 bits. Para estas 25 interconexões defeituosas, a melhor situação consiste em utilizar roteamento adaptativo, já que ele oferece um menor impacto.

O pior caso de falha é considerado quando as falhas atingem mais de 50% das interconexões, tornando as interconexões inutilizáveis para algumas situações conforme já mostrado neste trabalho (situações que o roteamento adaptativo não pode lidar). Porém, para que mais de 50% de uma interconexão seja defeituosa a probabilidade é de 1/72 para cada interconexão em uma rede intra-chip 4x3, e 1/384 para uma rede 8x8. Para a rede 8x8 com 100 fios defeituosos distribuídos pela rede, o pior cenário mostra

uma perda de 30% de conectividade para nossa proposta, enquanto que (KAKOEE et al., 2011) tem quase 70% de perda na conectividade.

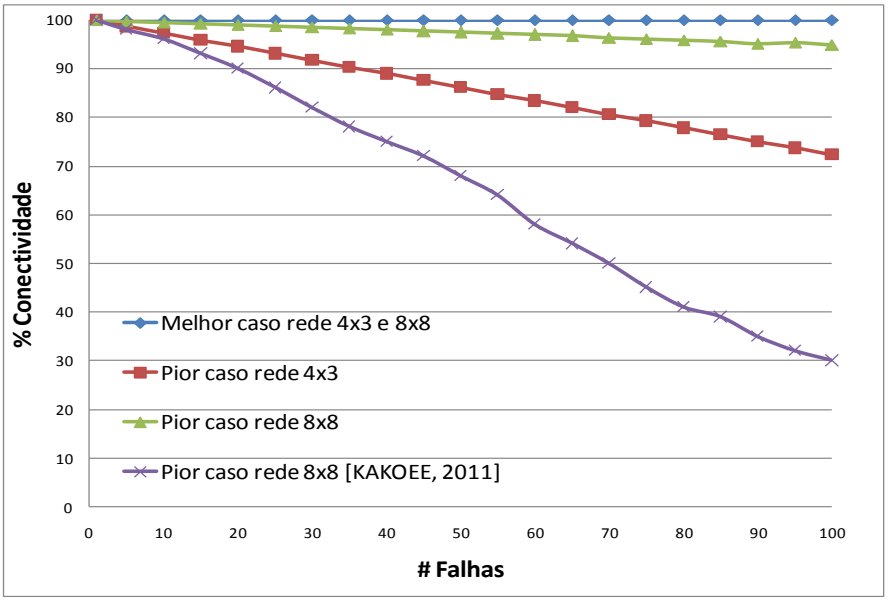


Figura 10.1: Conectividade da rede de acordo com o número de falhas nas interconexões.

Quando comparamos a conectividade entre as abordagens utilizadas neste trabalho, podemos aproximar os resultados conforme apresentado na figura 10.2 para o caso em que o MPEG4 é considerado, com 8 bits de largura de canal. Para tal situação, a distribuição de falhas foi realizada da melhor forma que cada proposta pode lidar. Para o código de Hamming, a melhor distribuição de falhas acontece quando existe até 1 fio defeituoso em cada interconexão, porém, quando mais de 1 falha acontece em alguma interconexão, a conectividade começa a reduzir, então nós consideramos o melhor caso quando todas interconexões sofrem com uma falha única para que a conectividade seja mantida em 100%. Quando mais de 72 falhas acontecem na rede, então a conectividade começa a reduzir, mesmo considerando que cada interconexão defeituosa terá 100% de falhas, e que as falhas irão se espalhar pelas outras interconexões gradativamente. Para a proposta que considera apenas o roteamento adaptativo executando na rede (RRADD – sem DD ativo) significa que múltiplas falhas foram consideradas dentro de cada fio, para fazer uso do melhor caso com roteamento adaptativo. Então o RA pode tolerar até em torno de 10% de interconexões completamente defeituosas, para então começar a ter dificuldades de exercer o roteamento adequado, tornando-se impossibilitado de evitar a perda de conectividade. A estratégia que utiliza a divisão de dados considera que o melhor caso de distribuição das falhas acontece quando até 50% dos fios de uma interconexão estão defeituosos. Então a seguinte distribuição foi considerada: após 50% de defeitos em todos os fios de cada interconexão, a perda na conectividade vai acontecendo conforme cada link vai tornando-se completamente inutilizável (vamos considerar que cada link vai recebendo os defeitos gradualmente, e apenas após um link ter 100% de fios defeituosos é que outro link começa a ter mais de 50% de fios

defeituosos). Para o caso em que a divisão de dados é combinada com o roteamento adaptativo, a estratégia consegue uma margem de tolerância a falhas um pouco maior, uma vez que o roteamento adaptativo permite ainda encontrar outro caminho para a comunicação, mesmo quando existem caminhos que não podem mais utilizar a divisão de dados. Para este cenário, não estamos levando em conta a quantidade de interconexões que não é utilizada na rede, fator este que permite aumentar a conectividade desde que as falhas estejam localizadas dentro das interconexões não utilizadas.

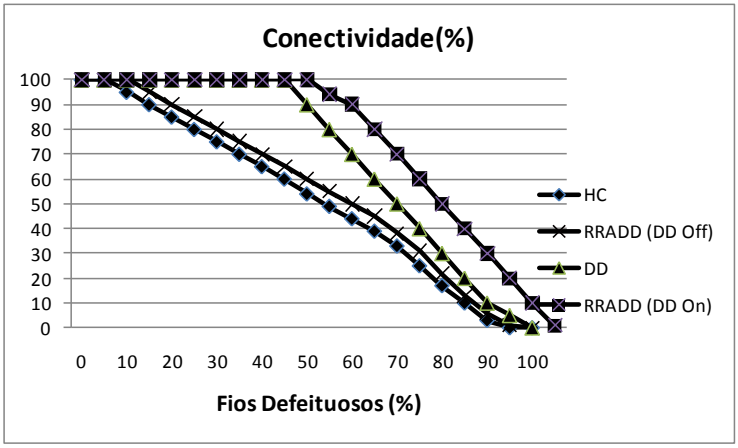


Figure 10.2: Análise da conectividade para as estratégias analisadas neste trabalho.

A figura 10.3 exibe a melhor configuração de falhas para cada estratégia. Para RRADD com DD desativado, a melhor configuração acontece quando todas as falhas estão concentradas dentro de poucas interconexões, para garantir a conectividade utilizando apenas o roteamento adaptativo. O pior caso para Hamming surge quando mais de uma falha acontece em cada interconexão. Mas para o RRADD com DD ativo, até 50% de fios podem estar defeituosos dentro de cada interconexão que a conectividade ainda poderá ser mantida, para o caso de 8 bits. Para 32 bits, a quantidade de fios defeituosos cai para 18,75%, embora ainda seja um valor aceitável.

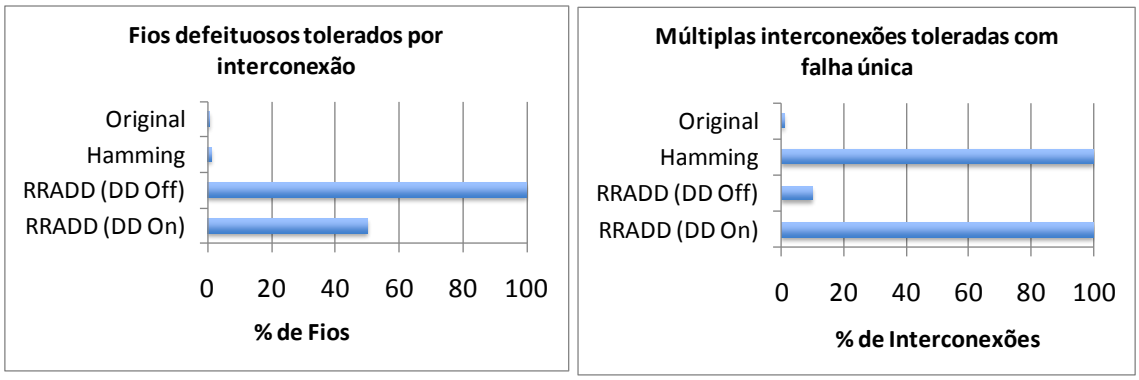


Figura 10.3: Capacidade de cada estratégia para lidar com múltiplos fios defeituosos.

11. CONCLUSÕES E TRABALHOS FUTUROS

As redes intra-chip estão tornando-se cada vez mais comuns na literatura como solução para prover a comunicação entre muitos núcleos dentro de um chip. Porém, diversos tipos de falhas podem aparecer num circuito, impedindo um bom funcionamento da rede, tornando-a inadequada para exercer determinadas aplicações. Consequentemente, diversas soluções precisam ser desenvolvidas para tolerar diferentes tipos de falhas. Neste trabalho, o foco principal consiste em tolerar múltiplas falhas permanentes em cada interconexão e em múltiplas interconexões, permitindo que as interconexões utilizem apenas os fios livres de falhas para prover a comunicação, tolerando de 50 a 18,75% de fios defeituosos para arquiteturas com 8 e 32 bits de canal, respectivamente, evitando criar regiões de isolamento devido à localização das falhas.

Para prover tolerância a falhas nas interconexões, foram utilizadas as técnicas de roteamento adaptativo e divisão de dados, e ambas as técnicas puderam ser combinadas com o remapeamento das tarefas para minimizar o impacto da falha no tempo de computação das aplicações, de forma a minimizar também a energia dissipada pela rede. A técnica de roteamento adaptativo possui mínima área extra e mínimo impacto na frequência, e por isso é escolhida primeiramente para aplicar a tolerância na rede. Já a divisão de dados tem um impacto variável, e pode penalizar o tempo de cada comunicação em até duas vezes o tempo original no caso da presença de falhas. Já com uma segunda abordagem da divisão de dados, que incluí a utilização de *latches*, o impacto pode ser reduzido, pois a informação é dividida e enviada dentro de um mesmo ciclo de relógio, sem qualquer prejuízo ao tempo de comunicação. De qualquer forma, a divisão de dados somente é utilizada quando o roteamento adaptativo não pode lidar com a situação, o que minimiza em geral os impactos que ela pode causar na rede.

O grande diferencial da proposta apresentada neste trabalho em relação aos trabalhos propostos na literatura é que nenhum fio extra foi adicionado nas interconexões. A quantidade de fios tornou-se um fator bastante crítico para a produção de circuitos, conforme apresentado na figura 11.1, uma vez que o atraso dos fios não tem reduzido na mesma proporção que o atraso das portas lógicas. Para adaptar um roteador original foram necessárias algumas alterações para permitir o roteamento adaptativo e a inserção de blocos com multiplexadores para a escolha dos fios corretos, no caso de situações que não possam ser solucionadas pelo roteamento adaptativo, caracterizando a divisão de dados. Para minimizar a energia, esses blocos são utilizados apenas nos roteadores que possuem falhas nas interconexões ligadas diretamente a eles.

O uso do remapeamento, da forma específica como foi apresentado ao longo deste trabalho, não agrega nenhuma penalidade para a rede em determinadas circunstâncias (redes homogêneas), e pode fazer com que a falha permaneça numa região não utilizada pela rede intra-chip, tornando o efeito da falha nulo, já que a interconexão não será utilizada.

Nossa proposta mostra que pode proteger a rede contra múltiplas falhas nas interconexões e também contra múltiplas interconexões defeituosas, com apenas 28% de área extra para uma tecnologia de 90 nm, além de mostrar que o impacto no tempo de computação de uma aplicação é variável de acordo com o tipo de falha e a sua localização, quando a primeira abordagem é utilizada. Para quase todas as situações, nossa proposta apresentou bons resultados após o remapeamento, e apenas situações que possuem a comunicação muito similar entras as regiões da rede (tráfego sintético complementar) é que nossa proposta não conseguiu obter melhores resultados que a solução utilizando Hamming, embora nossa cobertura de falhas seja superior. Para os benchmarks MPEG4, VOPD e para o tráfego sintético borboleta foi possível constatar que os resultados finais podem superar a solução com Hamming em aspectos de energia e tempo de computação, além de proteger a rede contra múltiplas falhas. Para o H.264 a mesma frequência foi considerada (300 MHz) para nossa proposta e para o Hamming e por isso não foi possível obter melhorias significativas para nossa abordagem neste caso em termos de tempo. Porém, quando a abordagem proposta insere *latches* na implementação, é possível transmitir os dados na interconexão duas vezes mais rápido que a frequência máxima do roteador, e foi possível constatar que todos os resultados podem superar a abordagem com o Hamming, mostrando no máximo 39% de impacto na potência e energia, enquanto que o Hamming pode ter até 54% de impacto mesmo quando apenas poucas falhas são diagnosticadas na rede (considerando apenas uma falha por interconexão). Além disso, o trabalho proposto pode permitir uma alta conectividade na rede de acordo com a localização das falhas, independente da utilização dos *latches*.

11.1 Trabalhos Futuros

Como trabalhos futuros, o remapeamento de tarefas requer um pouco mais de pesquisa e aprimoramento, uma vez que precisamos de uma solução eficiente para redes com núcleos heterogêneos. Na literatura, por exemplo, existe um trabalho que utiliza remapeamento das tarefas em FPGA (SINGH et al., 2007), mas cita que é necessário aprimorar como trabalho futuro o fato de incluir o suporte para plataformas heterogêneas. Na abordagem que foi apresentada ao longo deste trabalho, as estimativas de remapeamento foram realizadas com base na possibilidade da troca de posições dos núcleos, considerando que a rede intra-chip é composta por processadores, o que não retrata necessariamente a realidade.

Também se pretende analisar o impacto que ocorre na rede quando mais fios são acrescentados. Até então, o impacto considerado foi o número total de fios existentes e a potência dissipada pelos mesmos, de acordo com o comprimento estimado dos fios, mas pretende-se avaliar qual o impacto em termos de geração de falhas, como por exemplo, estudar se o aumento da quantidade de fios implica em ter um aumento proporcional ao número de falhas, uma vez que as falhas podem estar ligadas a uma chance maior de acontecerem curtos ou circuitos abertos com o aumento das interconexões. Avaliar o impacto dos fios em termos de área ocupada e impacto na temporização do projeto também é um trabalho que se pretende desenvolver posteriormente com mais detalhes.

Além disso, sabe-se atualmente que a interconexão tem se tornado limitante para a frequência em circuitos produzidos com tecnologias mais recentes. Como mostra o gráfico da figura 11.1, o atraso relativo das interconexões aumenta com o avanço da tecnologia enquanto o atraso da lógica diminui, e por isso será necessário pensar em novas soluções em que o uso de muitos fios conectando multiplexadores não seja o gargalo fundamental do projeto. Para as tecnologias utilizadas de 90 nm e 65 nm, o problema ainda não aparece claramente manifestado, mas para tecnologias mais recentes pode ser que o trabalho proposto não tenha bons resultados, uma vez que os fios limitarão a frequência da rede. Fios longos podem não ser uma boa solução no futuro, e precisam ser tratados com a devida atenção.

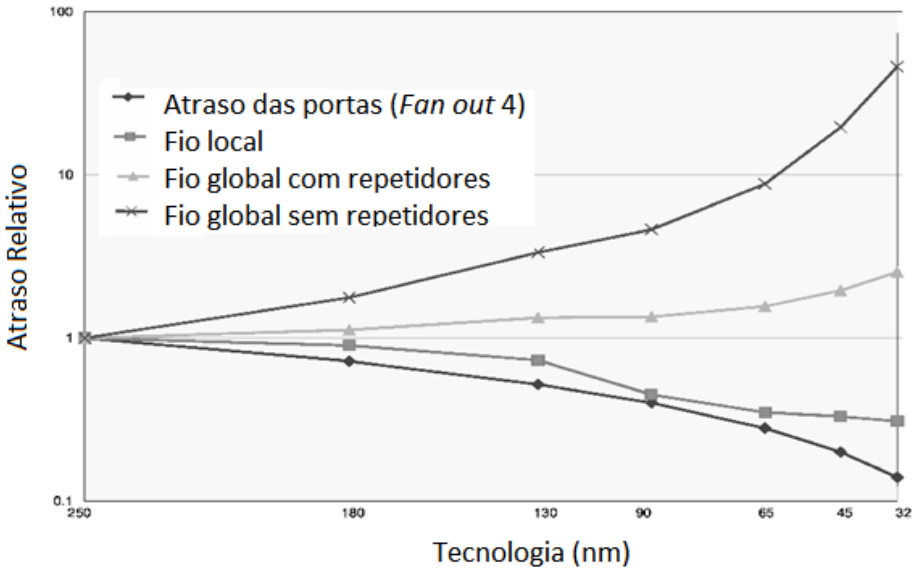


Figura 11.1: Atraso relativo entre os fios e a lógica de um circuito de acordo com a tecnologia utilizada (ITRS, 2009).

Por fim, pretende-se desenvolver posteriormente uma última abordagem para encontrar a melhor relação entre o emprego das técnicas apresentadas: roteamento adaptativo, divisão de dados e remapeamento. Com essa abordagem, será possível decidir qual será a primeira estratégia a ser adotada para a tolerância a falhas de forma a melhorar ainda mais o custo-benefício para a rede intra-chip, de acordo com a ordem do

emprego das técnicas estudadas. Neste trabalho, o remapeamento foi considerado posterior à inserção das técnicas de roteamento adaptativo e divisão de dados, e apenas para situações de falhas nas interconexões entre núcleos e roteadores. Mas o remapeamento também pode ser empregado para interconexões apenas entre roteadores, de modo que a interconexão defeituosa seja mapeada para uma região que não será utilizada pela rede, evitando assim o uso do roteamento adaptativo. Desta forma, obtém-se um impacto nulo da falha, e por isso é que se pretende desenvolver um mecanismo que escolha qual técnica empregar primeiro, de acordo com o melhor resultado a ser obtido, visando melhores benefícios para a rede. Outro aspecto a ser avaliado também é o caso da temporização que diz respeito aos *latches* incluídos na proposta, para avaliar se existe algum problema relativo a possíveis casos de metaestabilidade.

REFERÊNCIAS

ANDREWS, J. and BAKER, N. Xbox 360 System Architecture. IEEE Micro, Vol. 26. no. 2, pp. 25-37, 2006.

BANERJEE K. et al.. 3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration.Proceedings of the IEEE, 602-633, 2001.

BENINI, L. and DE MICHELI, G. Network on Chips: A new SoC Paradigm. IEEE Computer, p.70-78, 2002.

BERTOZZI, D. and BENINI, L. Xpipes: a network-on-chip architecture for gigascale systems-on-chip. IEEE Circuits and Systems Magazine, vol.4, no.2, pp. 18- 31, 2004.

BERTOZZI, D. et al. NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip. IEEE Transaction on Parallel and Distributed System, 113-129, 2005.

BRAGA M., et al. Efficiently using data Splitting and Retrasmission to Tolerante Faults in Networks-on-Chip Interconnects. IEEE International Symposium on Circuits and Systems (ISCAS), 2010.

CARRO, L. Projeto e prototipação de sistemas digitais. Porto Alegre, Brasil: Ed.Universidade/UFRGS, 2001.

CHIU, G. The odd-even turn model for adaptive routing. IEEE Transactions onParallel and Distributed Systems, vol.11, no.7, pp.729-738, 2000.

CHOUDHURY, A., et al. Yield Enhancement by Robust Application-specific Mapping on Network-on-Chips. Second International Workshop on Network on-Chip Architectures (NoCArc'09), pp. 37-42, 2009.

COLINGE, Jean-Pierre. FinFETs and Other Multi-Gate Transistors, Springer, New York, 2008.

CONCATTO C. et al. NoC Power Optimization Using a Reconfigurable Router. *IEEE Computer Society Annual Symposium on VLSI*, pp. 235-240, 2009.

CONCATTO, C. et al. Improving yield of torus NoCs through fault-diagnosis-and-repair of interconnect faults. 15th IEEE International On-Line Testing Symposium (IOLTS), pp.61-66, 2009.

CONCATTO, C. et al. Coping with Permanent Faults in NoCs by using Adaptive Strategies based on Router Design-level and Routing Algorithm-level. Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Microeletrônica. Porto Alegre, 2009.

CUVIELLO, M. et al. Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects. Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, pp. 297-303, 1999.

DEHON, A. and NAEIMI, H. Seven strategies for tolerating highly defective fabrication. IEEE Design & Test of Computers,vol.22, no.4, pp. 306- 315,2005.

DUATO, J. et al. Interconnection Networks, an Engineering Approach. IEEE Computers Society Press, 1997.

FRANTZ A., et al. Dependable Network-on-Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk. Proceedings International Test Conference (ITC), vol. 1, pp. 1 – 9, 2006.

FURBER S. Living with Failure: Lessons from Nature? Proceedings of the Eleventh IEEE European Test Symposium (ETS'06) - Vol.00, p.4-8, 2006.

GANGULY A., et al. Crosstalk-Aware Channel Coding Schemes for Energy Efficient and Reliable NoC Interconnects. IEEE Transactions on VLSI (TVLSI) Vol. 17, No.11, pp. 1626-1639, 2009.

GLESNER, M. Nanoelectronics Frontiers: A Research Agenda for Global Collaboration. Workshop Brasil-Alemanha de Micro e Nanoeletrônica, Porto Alegre, 2010.

GOOSSENS, K. and HANSON, A. The aethereal network on chip after ten years: Goals, evolution, lessons, and future. 47th ACM/IEEEDesign Automation Conference (DAC'10), pp.306-311, 2010.

GUERRIER, P. and GREINER, A. A generic architecture for on-chip packet-switched interconnections. Design, Automation and Test in Europe Conference and Exhibition (DATE'00), pp.250-256, 2000.

HERVÉ, M. et al. Diagnosis of interconnect shorts in mesh NoCs. 3rd ACM/IEEE International Symposium on Networks-on-Chip, 256-265, 2009.

ITRS, International Technology Roadmap for Semiconductors - Interconnect, 2009.

HOSSEINABADY, M. and NUNEZ-YANEZ, J.; "Fault-tolerant dynamically reconfigurable NoC-based SoC," International Conference on Architectures and Processors Application-Specific Systems, pp.31-36, 2008.

JEANG Y. et al. A Methodology Based on Maximal-Profit Spanning Tree for Designing Application Specific Networks on Chip (ASNOC). First International Conference onInnovative Computing, Information and Control, (ICICIC'06), vol.2, pp.18-21, 2006.

JINGCAO, H. et al. System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design. In IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems. pp. 2919-2933, 2006.

KAKOEE, M. et al. A Reliable Network for Priority-Based On-Chip Communication. Design, Automation & Test in Europe (DATE'11). Grenoble, France, 2011.

KOIBUCHI M., et al. A Lightweight Fault-Tolerant Mechanism for Network-on-Chip. Second ACM/ IEEE International Symposium on Networks-on-Chip, pp. 13-22, 2008.

KOLOGESKI, A. et al. Adaptive Buffer Size Based on Flow Control Observability for NoC Routers. XXV SIM - South Symposium on Microelectronics. Porto Alegre, Brasil, 2010.

KOLOGESKI, A. et al. Adaptive Approach to Tolerate Multiple Faulty Links in Network-on-Chip. In 12th IEEE Latin American Test Workshop (LATW2011). Porto de Galinhas, Brazil, 2011.

KOLOGESKI, A. et al. Improving Reliability in NoCs by Application-Specific Mapping Combined with Adaptive Fault-Tolerant Method in the Links. In 16th IEEE European Test Symposium (ETS'11). Trondheim, Norway, 2011.

LANGEN, D et al. High level estimation of the area and power consumption of on-chip interconnects. International ASIC/SOC Conference, 2000, Arlington, pp. 297-301, 2000.

LAN, Y. et al. BiNoC: A bidirectional NoC architecture with dynamic self-reconfigurable channel. . 3rd ACM/IEEE International Symposium onNetworks-on-Chi (NoCS'09)., pp.266-275, 2009.

LEHTONEN, T., et al. Online Reconfigurable Self-Timed Links for Fault Tolerant NoCs. IEEE International VLSI Design, 2007.

LONG C. and HE L.. Distributed sleep transistor network for power reduction. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 937-946, 2004.

MANDELLI M., et al. Multi-Task Dynamic Mapping onto NoC-based MPSoCs. 24th Symposium on Integrated Circuits and Systems Design (SBCCI), João Pessoa, Brazil, 2011.

MELO, A. et al. Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC. 18th Symposium on Integrated Circuits and Systems Design (SBCCI'05). New York, USA: ACM Press, pp. 178-183, 2005.

MELO, A. Qualidade de Serviço em Redes Intra-Chip: Implementação e avaliação sobre a Rede Hermes. Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação. Porto Alegre, 2006.

MOORE, GORDON E. Cramming more components onto integrated circuits. Electronics Magazine, 1965.

MURALI S. Designing Reliable and Efficient Networks on Chips. New York, USA. Springer, 2009.

NICOPOLOUS, C. et al. ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers. 39th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 333-346, 2006.

NICOPOLOUS C., et al. Network-on-Chip Architectures: A Holistic Design Exploration. New York, USA. Springer, 2009.

OST L., et al. Exploring Dynamic Mapping Impact on NoC-based MPSoCs Performance Using a Model-based Framework. 24th Symposium on Integrated Circuits and Systems Design (SBCCI), João Pessoa, Brazil, 2011.

PALERMO, G. et al. Application-Specific Topology Design Customization for STNoC. 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp.547-550, 2007.

PALESI, M., et al. Leveraging Partially Faulty Links Usage for Enhancing Yield and Performance in Networks-on-Chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.29, no.3, pp.426-440, 2010.

RABAEY, J. et al. Digital Integrated Circuits: A Design Perspective, 2 ed. Upper Saddle River, NJ: Prentice Hall/Pearson Education, 2003.

REIS, R. Concepção de Circuitos Integrados. Porto Alegre, Brasil: Ed. Sagra Luzatto, 2000.

SAKURAI T., Approximation of wiring delay in MOSFET LSI, IEEE Journal of Solid-State Circuits, Vol.18, 1983.

SCHONWALD T., et al. Fully Adaptative Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures. 10th Euromicro Conference on Digital System Design Architecture, Methods and Tools, pp. 527-534, 2007.

SHI K. and HOWARD D.. Sleep Transistor Design and Implementation - Simple Concepts Yet Challenges To Be Optimum. International Symposium on VLSI Design, Automation and Test, 2006.

SINGH, A.K. et al. Mapping real-life applications on run-time reconfigurable NoC-based MPSoC on FPGA.International Conference on Field-Programmable Technology (FPT), pp.365-368, 2010.

STALLINGS, W. Data and computer communications. 8th ed. Upper Saddle River: Pearson Prentice Hall, 2007.

STENSGAARD, M. and SPARSO, J.. ReNoC: A Network-on-Chip Architecture with Reconfigurable Topology. Second ACM/IEEE International Symposium on Networks-on-Chip, pp.55-64, 2008.

TORNERO, R., et al. A multi-objective strategy for concurrent mapping and routing in networks on chip. IEEE International Symposium on Parallel&Distributed Processing, pp.1-8, 2009.

VENKATARAMAN S. and DRUMMONDS S.B.. "POIROT: a Logic Fault Diagnosis Tool and its Applications," International *Test Conference,* pp.253-262, 2000.

VU-DUC NGO, et al. Analyzing the Performance of Mesh and Fat-Tree topologies for Network on Chip design. LNCS (Springer-Verlag), pp 300-310, 2005.

YANG, H. and PAPACHRISTOU, C. A Method for Detecting Interconnect DSM Defects in Systems on Chip, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.25, no. 1, pp.197-204, 2006.

ZEFERINO, C. Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho. Tese submetida à avaliação como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação. Porto Alegre, 2003.

ZEFERINO C. and SUSIN A. SoCIN: A Parametric and Scalable Network-on-Chip, 16th Symposium on Integrated Circuits and System Design (SBCCI). São Paulo, Brazil, pp. 169-174, 2003.

ZEFERINO, C. et al. RASoC: a router soft-core for networks-on-chip. Design, Automation and Test in Europe Conference and Exhibition (DATE'04), vol.3, pp. 198-203, 2004.

ZHANG W. et al. Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip. *Global Congress onIntelligent Systems (GCIS '09)* - vol.3, pp.329-333, 2009.

ANEXO I: ARTIGOS PUBLICADOS

Durante o desenvolvimento deste trabalho, muitas publicações foram desenvolvidas. O primeiro trabalho, intitulado "*Adaptive Router Architecture Based on Traffic Behavior Observability*" publicado em dezembro de 2009, corresponde a sequência de um trabalho que já estava em andamento, em que eu pude participar desenvolvendo um monitor de tráfego para a rede, a fim de permitir a adaptabilidade dos recursos de buffer de acordo com o tráfego de cada canal. O segundo trabalho, "*Adaptive Buffer Size Based on Flow Control Observability for NoC Routers*", publicado em maio de 2010 em Porto Alegre, no Simpósio Sul de Microeletrônica, foi um breve resumo do primeiro trabalho, bem como "*Monitor-Adapter Coupling for NOC Performance Tuning*" em julho do mesmo ano.

Em junho de 2010, o trabalho "*Run Time Adaptive Mechanism for Sustaining Performance with Fault Tolerance in NOCs*" abordou a adaptabilidade dos buffers a fim de melhorar a performance da rede através do empréstimo de buffers e por tolerar buffers defeituosos de um roteador.

Já o trabalho publicado no LATW, em Porto de Galinhas, no ano de 2011, foi o primeiro trabalho publicado que abordou o assunto desta dissertação. Foram introduzidas as técnicas de roteamento adaptativo e divisão de dados a fim de tolerar falhas nas interconexões da rede intra-chip, porém a possibilidade do uso de remapeamento não foi citada neste trabalho. Na sequência, um trabalho mais completo sobre o assunto foi publicado no ETS 2011, no mês de maio, na Noruega, abordando o remapeamento dos núcleos.

Recentemente, nosso grupo também teve um trabalho aceito para a revista MICPRO, que é uma extensão do trabalho sobre a adaptabilidade nos buffers combinada com tolerância a falhas, porém mais completo e detalhado.

Os dois últimos trabalhos publicados são bastante recentes. O trabalho publicado em setembro na cidade de João Pessoa, na conferência denominada SBCCI, fala sobre topologia adaptativa, que se divide entre uma rede com topologia grelha e uma rede com topologia irregular, semelhante a uma árvore. Este trabalho foi desenvolvido numa das disciplinas do mestrado. O outro trabalho, publicado em setembro no VLSI-SoC, em Hong Kong, é fruto de uma colaboração minha com o Politécnico de Milão, durante

os meses de Fevereiro a Abril de 2011, e consiste no desenvolvimento de um simulador para medir a latência e o *throughput* dos dados numa rede com adaptabilidade nos buffers e nos canais virtuais, bem como permite também a comparação com um trabalho amplamente conhecido na literatura (ViChaR), proposto por (NICOPOLOUS et al., 2006).

Por fim, um último trabalho foi recentemente submetido para a revista *IEEE Design & Test* e aguarda pela avaliação. Esta submissão mostra em detalhes a estratégia da inserção dos *latches* discutida neste trabalho.

Todos os trabalhos citados são apresentados na tabela A como consta a seguir.

Tabela A: Trabalhos publicados durante o mestrado acadêmico.

Publicações		
Titulo do Artigo	**Veículo**	**Status**
Adaptive Router Architecture Based on Traffic Behavior Observability	*Workshop* Internacional - NoCArc 2009	Aceito
Adaptive Buffer Size Based on Flow Control Observability for NoC Routers	Simpósio Nacional - SIM 2010	Aceito
Monitor-Adapter Coupling for NOC Performance Tuning.	Conferência Internacional - SAMOS X 2010	Aceito
Run Time Adaptive Mechanism for Sustaining Performance with Fault Tolerance in NOCs	*Workshop* Internacional - DSNoC 2010	Aceito
Adaptive Approach to Tolerate Multiple Faulty Links in Network-on-Chip	Conferência Internacional - LATW 2011	Aceito
Improving Reliability in NoCs by Application-Specific Mapping Combined with Adaptive Fault-Tolerant Method in the Links	Conferência Internacional - ETS 2011	Aceito
A NoC Closed-Loop Performance Monitor and Adapter	Revista MICPRO 2011	Aceito
AdNoC Case-Study for Mpeg4 Benchmark: Improving Performance and Saving Energy with an Adaptive NoC	Simpósio Internacional - SBCCI 2011	Aceito
Two-Levels of Adaptive Buffer for Virtual Channel Router in NoCs	Conferência Internacional - VLSI-SOC 2011	Aceito
ATARDS: An Adaptive Fault-Tolerant Technique to Cope with Massive Defects in Network-on-Chip Interconnections	IEEE Design & Test Magazine	Submetido