

Análise de Desempenho de um Roteador utilizando Diferentes Arquiteturas de Decremento em uma Rede-em-*Chip*

T. de OLIVEIRA¹, Departamento de Engenharia Elétrica, FEIS, UNESP, 15385-000
Ilha Solteira, SP, Brasil.

N. MARRANGHELLO², Departamento de Ciências de Computação e Estatística,
IBILCE, UNESP, 15054-000 São José do Rio Preto, SP, Brasil.

Resumo. Uma rede-em-*chip* está sendo desenvolvida para permitir a implementação de uma Rede de Petri em *hardware*. Para determinar a melhor arquitetura do decrementador a ser incorporada ao roteador dessa rede foi desenvolvida uma abordagem baseada em equações matemáticas que computam as quantidades de portas lógicas e de níveis de lógica dos decrementadores e do roteador. Uma fórmula de desempenho foi estabelecida para realizar uma análise comparativa da arquitetura do roteador com cada um dos decrementadores.

Palavras-chave. Roteador, decrementadores, desempenho.

1. Introdução

Devido a um mercado atual bastante competitivo, o projeto de sistemas digitais tem se tornado cada vez mais complexo e exigindo um tempo de confecção bastante curto. Diante deste panorama, vem surgindo uma necessidade crescente de ferramentas CAD que possam auxiliar os projetistas na tomada de decisão durante o desenvolvimento de um sistema. Análises realizadas antes da etapa de implementação física podem economizar custos e melhorar o desempenho do sistema em relação ao consumo de energia, tempo de resposta entre outros fatores. Com isso, algumas técnicas têm sido propostas para a estimativa da área, tempo de resposta e consumo de energia de um dado sistema digital. Ramachandran et al. [6] realizam uma estimativa da área e do tempo de resposta de um sistema baseando-se em modelos com arquiteturas de *layouts* pré-definidas. Srinivasan et al. [7] estimam a área e o tempo de resposta total do sistema utilizando informações provenientes da síntese lógica de apenas um subconjunto de circuitos do projeto. Nemani e Najm [5] realizam uma estimativa da área e do consumo de energia baseando-se na quantidade e no tamanho dos implicantes primos das funções booleanas que compõem

¹tiagoooli@yahoo.com.br

²norian@ibilce.unesp.br

o sistema. Büyüksahin e Najm [1] descrevem o sistema por meio de uma rede booleana e, a extração de determinadas características dessa rede, como a quantidade de nós, a quantidade de arcos e o grau dos nós permite uma estimativa da área do sistema. Gelosh e Steliff [3] utilizam técnicas de aprendizado de máquina para modelar a própria ferramenta de síntese e utilizar esse modelo para a obtenção da área e do tempo de resposta do sistema.

Imagine-se um sistema digital composto por vários blocos digitais, precisando executar uma determinada tarefa a qual pode ser implementada por meio de arquiteturas diferentes. Este artigo procura identificar, para o projeto de um roteador numa rede-em-*chip*, a melhor arquitetura que deve ser implementada em cada bloco funcional para que a rede-em-*chip* como um todo atinja o seu desempenho máximo. Além disso, numa possível expansão dos sinais de entrada/saída do roteador, essas equações matemáticas podem ser reutilizadas para determinar a configuração que deve ser adotada para se obter o desempenho estabelecido.

2. Motivação deste Trabalho

Baseando-se nos conceitos de rede-em-*chip* [4], nosso grupo de projeto desenvolveu em VHDL um roteador capaz de determinar o trajeto de um pacote de dados sobre uma rede-em-*chip* com uma topologia de malha bidimensional. Na arquitetura do roteador projetado foi necessário o uso de estruturas aritméticas (decrementadores) para computar o endereço de destino de um pacote. A rede-em-*chip* desenvolvida recebe pacotes contendo, em seu cabeçalho, alguns campos referentes ao endereço de destino (variações nos eixos x e y). Ao receber um pacote, o roteador verifica os valores armazenados nos campos de variações. Se forem zeros, o pacote chegou a seu destino. Caso contrário, um campo de variação é decrementado em uma unidade e o pacote é enviado a um roteador vizinho.

Arquiteturas diferentes podem ser utilizadas nesses decrementadores levando em consideração a forma de obtenção do bit de transporte. Dependendo da arquitetura utilizada para esses decrementadores o roteador pode alcançar uma velocidade maior de processamento ou/e uma área menor numa possível implementação física da arquitetura proposta num *chip* ou num dispositivo lógico programável. Uma maior velocidade de execução do sistema de comunicação, composto por um conjunto de roteadores, diminuirá o tempo de resposta da rede-em-*chip* proposta. Por sua vez, a redução da área de cada roteador pode reduzir o custo da fabricação em um circuito integrado da rede proposta.

A abordagem apresentada neste artigo tem como propósito comparar o desempenho do roteador com as principais e mais conceituadas técnicas de obtenção do sinal de transporte, quais sejam, decrementadores com transporte em cascata, decrementadores com transporte antecipado e decrementadores com transporte selecionado e identificar a melhor técnica que deve ser utilizada no projeto do roteador.

Na próxima seção, especificam-se os conceitos fundamentais e as equações matemáticas que possibilitam a comparação de desempenho da arquitetura do roteador utilizando, como exemplo, o decrementador com transporte antecipado. Posteriormente, apresentam-se os resultados obtidos sobre o desempenho do roteador para cada decrementador, finalizando, com alguns comentários e discussões sobre os re-

sultados obtidos.

2.1. Decrementador com transporte antecipado

O decrementador com transporte antecipado [2] é capaz de determinar os bits de transporte de todos módulos de subtração utilizando apenas os sinais de entrada dos operandos (X e Y) e o vem-um inicial (C_0). Com isso, o cálculo dos bits de transporte é realizado simultaneamente, não necessitando realizar a propagação do vai-um como acontece no decrementador com transporte em cascata. A figura 1 mostra a arquitetura e o caminho crítico do decrementador com transporte antecipado. O circuito gerador de transporte antecipado determina os valores de todos os bits de transporte intermediários antes que os bits correspondentes da subtração sejam computados. O circuito gerador pode ser equacionado da seguinte forma

$$\begin{aligned}
 C_1 &= 0 + P_0 \\
 C_2 &= G_1 + 0 + P_1 \cdot P_0 \\
 &\vdots \\
 C_n &= G_{n-1} + P_{n-1} \cdot G_{n-2} + P_{n-1} \cdot P_{n-2} \cdot G_{n-3} + P_{n-1} \cdot P_{n-2} \cdot P_{n-3} \cdot G_{n-4} \\
 &\quad + \dots + P_{n-1} \cdot P_{n-2} \cdot P_{n-3} \dots P_2 \cdot G_1 + 0 + P_{n-1} \cdot P_{n-2} \dots P_0. \quad (2.1)
 \end{aligned}$$

Com relação à quantidade de níveis de lógica, pode ser observado que os sinais auxiliares P e G precisam de no máximo um nível de lógica, correspondente ao atraso de uma porta lógica *not*. Para a produção da subtração, de acordo com a fórmula $S_i = P_i \oplus C_i$, é necessário o atraso de mais uma porta lógica.

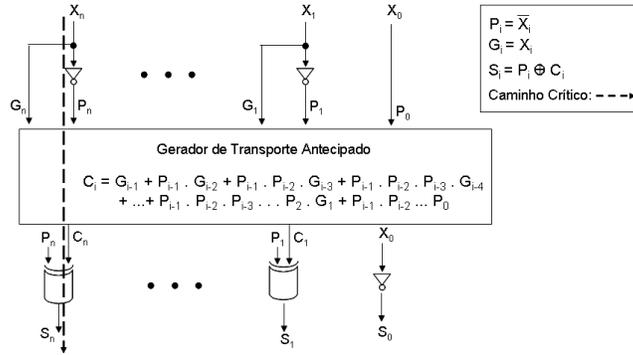


Figura 1: Caminho crítico para o decrementador com transporte antecipado

O circuito gerador de transporte antecipado realiza o cálculo dos sinais C_i em paralelo e, portanto, apenas o atraso do bit de transporte do último nível deve ser levado em consideração, visto que este sinal de transporte possui a expressão booleana que gera o maior atraso. Tomando $n + 1$ como o índice de maior nível, tem-se

$$C_{n+1} = G_n + P_n \cdot G_{n-1} + P_n \cdot P_{n-1} \cdot G_{n-2} + P_n \cdot P_{n-1} \cdot P_{n-2} \cdot G_{n-3} + \dots +$$

$$+P_n.P_{n-1}.P_{n-2} \dots P_2.G_1 + 0 + \underbrace{P_n.P_{n-1} \dots P_0}_{\text{maior atraso}}.$$

Note-se que para a geração do sinal C_{n+1} , os mintermos (termos de produtos, representado operações *and*) são realizados simultaneamente e, portanto, deve-se levar em consideração apenas o mintermo com maior atraso, ou seja, o mintermo que possuir um maior número de operandos. Na expressão anterior, o mintermo de maior atraso, possui $(n+1)$ entradas, que corresponde a um atraso de $\lceil \log_2(n+1) \rceil$ níveis de lógica. Após a execução da operação lógica *and*, deve-se computar o atraso máximo gerado pela realização das operações lógicas *or*, visto que os mintermos e as operações booleanas *or* são realizadas sequencialmente. Na expressão anterior para o sinal C_{n+1} , as operações lógicas *or* terão $(n+1)$ operandos de entrada, visto que $C_{n+1} = G_n + \dots + G_{n-1} + \dots + G_{n-2} + \dots + G_1 + P_n.P_{n-1} \dots P_0$. Portanto, para a realização das operações *or* será necessário um atraso máximo de $\lceil \log_2(n+1) \rceil$ níveis de lógica.

Assim, a fórmula completa indicando a quantidade de níveis de lógica da arquitetura do decrementador com transporte antecipado será

$$\begin{aligned} NL &= \lceil \log_2(n+1) \rceil + \lceil \log_2(n+1) \rceil + 2 \\ &= 2 * \lceil \log_2(n+1) \rceil + 2. \end{aligned}$$

Com relação à quantidade de portas lógicas do decrementador com transporte antecipado, primeiro, será calculada a quantidade de portas do gerador de transporte antecipado e depois, serão computadas as lógicas de produção dos sinais auxiliares P e G , e subtração. De acordo com a equação 2.1, a quantidade de portas lógicas (QP) necessárias para implementar a lógica do sinal C_1 é zero, para o sinal C_2 é 2 (uma operação lógica *and* entre P_1 e P_0 e uma operação lógica *or* entre os operandos G_1 e o resultado da operação *and*), assim $QP(C_1) = 0$, $QP(C_2) = 2$, $QP(C_3) = 5$, $QP(C_4) = 9$ e $QP(C_5) = 14$.

O sinal C_n possui n operandos para a realização da operação binária *or*, visto que:

$$C_n = \underbrace{G_{n-1} + \dots + G_{n-2} + \dots + G_{n-3} + \dots + G_{n-4} + \dots + G_1}_{n-1 \text{ operandos}} + P_{n-1} \dots P_0.$$

Assim, para a realização da operação *or* são necessárias $(n-1)$ portas lógicas *or* de duas entradas. O sinal C_n ainda possui $(n-1)$ termos contendo operações *and*, como mostrado na equação 2.1. A cada termo *and*, da esquerda para a direita da expressão de C_n , acrescenta-se um operando a mais, como mostra a tabela 1.

Assim, de acordo com a tabela 1, para a realização de todas as operações *and* em todos os termos do sinal C_n são necessárias $\sum_{i=1}^{n-1} i$ portas lógicas *and* de duas entradas. Desta forma, tem-se

$$QP(C_n) = \underbrace{(n-1)}_{\text{portas or}} + \underbrace{\sum_{i=1}^{n-1} i}_{\text{portas and}}.$$

Tabela 1: Acréscimo de portas lógicas de um termo para outro

Termo	Quantidade de portas	Correspondência
1	1	$P_{n-1}.G_{n-2}$
2	2	$P_{n-1}.P_{n-2}.G_{n-3}$
3	3	$P_{n-1}.P_{n-2}.P_{n-3}.G_{n-4}$
4	4	$P_{n-1}.P_{n-2}.P_{n-3}.P_{n-4}.G_{n-5}$
\vdots	\vdots	\vdots
$(n-1)$	$(n-1)$	$P_{n-1}.P_{n-2} \dots P_0$

A quantidade total de portas lógicas do gerador de transporte antecipado (GTA) pode então ser expressa da seguinte maneira:

$$QP(\text{GTA}) = \sum_{j=1}^n \overbrace{\left[(j-1) + \sum_{i=1}^{j-1} i \right]}^{QP(C_j)} .$$

$QP(C_1)+QP(C_2)+\dots+QP(C_n)$

Expandindo, tem-se $QP(\text{GTA}) = \sum_{j=1}^n \sum_{i=1}^{j-1} i + \sum_{j=1}^n j - n$.

Para finalizar o cálculo da quantidade de portas do decrementador com transporte antecipado, é necessário computar a quantidade de portas lógicas dos sinais auxiliares P e G , e subtração.

Como mostrado na figura 1, $P_0 = X_0$, $G_0 = 0$ e $S_0 = \overline{X_0}$ e assim, para a geração dos sinais auxiliares e subtração do primeiro bit, será necessário apenas uma porta lógica.

Para os demais P_i , G_i e S_i , tem-se $P_i = \overline{X_i}$, $G_i = X_i$ e $S_i = P_i \oplus C_i$. Assim, para cada $(P_i, G_i$ e $S_i)$ precisa-se de duas portas lógicas.

Portanto, tomando n como sendo o tamanho (quantidade de bits) do decrementador com transporte antecipado, a quantidade de portas lógicas para todos os sinais auxiliares e subtrações é dada por $QP(P, G$ e $S) = 1 + 2 * n$.

A quantidade de portas lógicas do decrementador com transporte antecipado é dada por: $QP = QP(\text{GTA}) + QP(P, G$ e $S)$. Substituindo, tem-se

$$QP = \left(\sum_{j=1}^n \sum_{i=1}^{j-1} i + \sum_{j=1}^n j - n \right) + (1 + 2 * n) .$$

3. Fórmula de Desempenho

A fórmula de desempenho envolvendo a quantidade de portas lógicas (QP) e a quantidade de níveis de lógica (NL) pode ser assim esquematizada

$$DE = P_{qp} * \left(\frac{QP_{min}}{QP} \right) + P_{nl} * \left(\frac{NL_{min}}{NL} \right) , \quad (3.1)$$

onde P_{qp} indica o peso dado ao desempenho em relação à quantidade de portas lógicas (QP) e P_{nl} indica o peso dado ao desempenho em relação à quantidade de níveis de lógica (NL). Os valores atribuídos aos pesos P_{qp} e P_{nl} devem satisfazer a seguinte relação: $P_{qp} + P_{nl} = 1$.

Esses pesos permitem um ajuste fino na análise matemática do sistema, o que pode permitir, por exemplo, que durante o processo de análise haja uma maior preocupação com a quantidade de níveis de lógica do sistema correspondente do que com a quantidade de portas lógicas resultante, ou vice-versa.

Embora a fórmula de desempenho 3.1 estabelecida leve em consideração apenas dois parâmetros, ela pode ser estendida para incluir parâmetros adicionais, como consumo de energia por exemplo.

A quantidade de portas lógicas pode ser utilizada para estimar o número de recursos necessários numa possível implementação física do sistema modelado. No caso de uma implementação em um circuito integrado, por exemplo, a quantidade de portas lógicas pode ser utilizada para estimar o tamanho da área que será ocupada pelo sistema, e por consequência, determinar o custo de sua fabricação.

Por sua vez, a quantidade de níveis de lógica pode ser utilizada para determinar a velocidade de operação do circuito. No caso do roteador, pode ser utilizada inclusive para definir a taxa teórica máxima de vazão de dados. O roteador possui cinco portas de comunicação de entrada/saída, envia e recebe pacotes compostos por 32 bits, assim, uma estimativa da taxa teórica máxima de vazão do roteador pode ser equacionada da seguinte forma: $5 * 32 * \text{velocidade de operação}$. Desta forma, as quantidades de portas lógicas e de níveis de lógica podem ser utilizadas para estimar importantes características do circuito implementado, como custo, velocidade de processamento e taxa de vazão.

4. Resultados

As fórmulas de desempenho e as fórmulas que indicam as quantidades de portas lógicas e de níveis de lógica dos decrementadores e do roteador foram descritas no MATLAB para a realização da análise comparativa da inserção desses decrementadores na arquitetura do roteador. A função desenvolvida gera dois gráficos referentes à comparação dos decrementadores. No primeiro gráfico, denominado Panorama 1, a função identifica o desempenho dos decrementadores sem levar em consideração a arquitetura do roteador na qual eles foram incluídos. A figura 2 mostra o gráfico do Panorama 1, considerando um peso de 70% em relação à quantidade de níveis de lógica, 30% em relação à quantidade de portas lógicas e um tamanho de bloco contendo 4 bits. Em alguns decrementadores analisados, vetores de 4 bits foram agrupados em blocos de decremento. Desta forma, pode-se construir decrementadores híbridos, utilizando uma técnica de transporte internamente nos blocos de decremento e outra técnica de transporte entre os blocos. Sete tipo de decrementadores com técnicas diferentes para a obtenção dos bits de transporte foram analisados, quais sejam: transporte em cascata, transporte antecipado, uma modificação do transporte antecipado, transporte em cascata nos blocos de decremento e transporte selecionado entre os blocos (transporte cascata-selecionado), transporte antecipado nos blocos e transporte selecionado entre os blocos (transporte

antecipado–selecionado), transporte antecipado modificado nos blocos e transporte selecionado entre os blocos (transporte antecipado modificado–selecionado) e, por fim, um decrementador com transporte antecipado nos blocos e transporte em cascata entre os blocos (transporte antecipado–cascata).

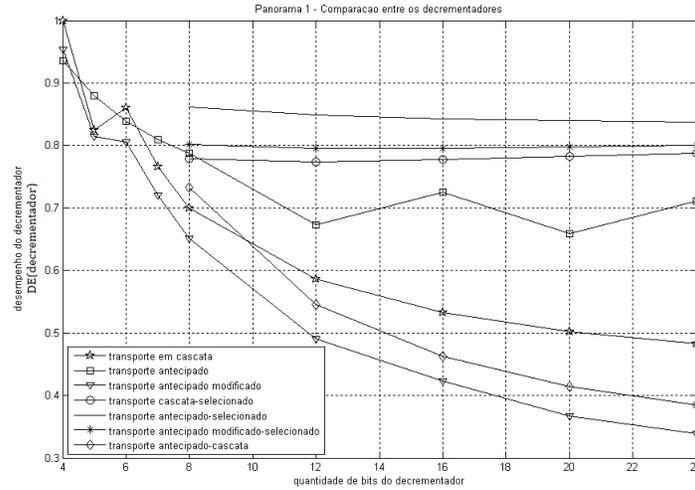


Figura 2: Panorama 1 – Comparação de desempenho entre os decrementadores

Pode-se notar que o decrementador com transporte antecipado–selecionado obteve o melhor desempenho entre 8 e 24 bits permanecendo na faixa de 0.8 e 0.9 de desempenho. Na faixa logo abaixo, entre 0.7 e 0.8 de desempenho, o segundo melhor desempenho foi do decrementador com transporte antecipado modificado–selecionado. O decrementador com transporte cascata–selecionado obteve a terceira melhor relação de desempenho. O pior desempenho foi do decrementador com transporte antecipado modificado atingindo um desempenho de aproximadamente 0.2 para 24 bits.

O segundo gráfico (Panorama 2), mostrado na figura 3, indica o desempenho da arquitetura do roteador projetada com cada um dos decrementadores.

Pode-se notar que o decrementador com transporte em cascata possui a melhor relação de desempenho entre 4 e 7 bits, obtendo o valor máximo, $DE = 1$. A partir de 8 bits, três decrementadores permaneceram na faixa de desempenho entre 0.95 e 1.00. Todos os três possuem como base a técnica de transporte selecionado entre os blocos, se diferenciando com relação à implementação interna do bloco. Assim, o decrementador com transporte cascata–selecionado obteve o melhor desempenho. Seguido de perto, pelo decrementador com transporte antecipado modificado–selecionado. O terceiro melhor desempenho é do decrementador com transporte antecipado–selecionado. O pior desempenho entre 8 e 24 bits ficou com o decrementador com transporte em cascata, atingindo a marca de aproximadamente 0.55 para 24 bits.

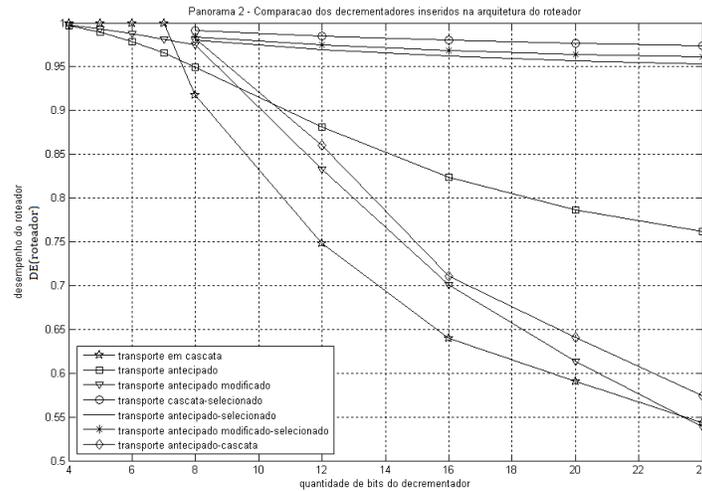


Figura 3: Panorama 2 – Desempenho do roteador com os decrementadores inseridos na arquitetura

5. Discussão

Como pode ser observado na figura 2, o decrementador com transporte antecipado-selecionado apresentou um desempenho melhor do que o decrementador com transporte cascata-selecionado. Porém, como mostra a figura 3, o roteador projetado com o decrementador com transporte cascata-selecionado obteve a melhor relação de desempenho, mesmo este decrementador tendo um desempenho individual inferior ao decrementador com transporte antecipado-selecionado.

Isso acontece porque na arquitetura do roteador, além de caminhos de dados existem também alguns caminhos relacionados ao controle da arquitetura. Com isso, a escolha de decrementadores mais rápidos pode melhorar o tempo de resposta de determinados caminhos de dados, mas contudo, pode não interferir no tempo de resposta dos caminhos de controle e de alguns caminhos de dados. Desta forma, apesar do decrementador com transporte antecipado-selecionado ser individualmente mais rápido, o desempenho em relação à quantidade de níveis de lógica do roteador projetado com qualquer um dos dois decrementadores será o *mesmo* pois o caminho crítico do roteador se mantém inalterado para os dois decrementadores. Contudo, o decrementador mais rápido que utiliza o transporte antecipado-selecionado gasta uma quantidade de portas lógicas um pouco maior do que o decrementador com transporte cascata-selecionado, e por este motivo a arquitetura do roteador utilizando o decrementador com transporte cascata-selecionado apresentou um desempenho melhor.

Também pode-se notar, na figura 3, que de modo geral os decrementadores que utilizam transporte selecionado entre os blocos apresentaram os melhores resultados. Convencionalmente, o somador com transporte selecionado tem um atraso relativamente pequeno, contudo, possui um número elevado de portas lógicas visto que,

em cada bloco, dois somadores são necessários para produzir o resultado. Porém, o somador com transporte selecionado foi adaptado para o projeto do roteador (realizando apenas o decremento de um) e esta adaptação permitiu a inclusão de apenas um somador (decrementador) em cada bloco, reduzindo significativamente o número de portas lógicas. Com isso, estes decrementadores além de serem rápidos passaram a ter uma pequena quantidade de portas lógicas, tornando-se extremamente atrativos.

A escolha de um peso de 70% em relação à quantidade de níveis de lógica e de 30% em relação à quantidade de portas lógicas se deve aos motivos explicados na seqüência.

A velocidade de operação dos roteadores se torna muito importante visto que pacotes gerados pela arquitetura poderão atravessar vários roteadores até chegarem aos seus destinos finais. Se a velocidade de operação for muito baixa, os pacotes levarão muito tempo para ser entregues às transições do sistema e, por consequência, o tempo de resposta da arquitetura poderá aumentar muito. Assim, para se conseguir uma arquitetura eficiente, os roteadores que compõem o sistema de comunicação devem ser os mais rápidos possíveis. Porém, a quantidade de lógica utilizada para o projeto de cada roteador influencia no tamanho do *chip* a ser construído, o que acarretaria um custo maior na fabricação dessa arquitetura. Além disso, roteadores com uma quantidade muito grande de lógica tomariam muito espaço no *chip*, o que diminuiria a quantidade de circuitos que poderiam ser colocados na implementação física dessa arquitetura. Com menos circuitos no *chip*, o algoritmo de mapeamento seria capaz de alocar uma quantidade menor de lugares e transições das redes de Petri.

Contudo, foram realizadas análises utilizando outros valores para os pesos em relação às quantidades de portas lógicas e de níveis de lógica. Uma função em MATLAB foi especialmente desenvolvida para retornar o desempenho do roteador com cada um dos decrementadores variando os pesos em relação à porta lógica de 0% até 100%, com uma taxa de incremento de 10%. Para a maioria dos pesos em relação à quantidade de portas lógicas, de 0 até 0.8, ou de 0% até 80%, os três decrementadores que utilizam transporte selecionado entre os blocos apresentaram os melhores resultados. A partir de um peso em relação à quantidade de portas lógicas de aproximadamente 87%, o decrementador com transporte em cascata ultrapassa o decrementador com transporte cascata-selecionado, se tornando o mais adequado. O decrementador com transporte em cascata possui uma quantidade reduzida de portas lógicas. Portanto, nesses casos, onde o nível de lógica não é quase levado em consideração, o decrementador com transporte em cascata leva maior vantagem apesar de apresentar o pior tempo de resposta.

6. Conclusão

Foram formuladas equações matemáticas que computam as quantidades de portas lógicas e de níveis de lógica de algumas estruturas aritméticas. Uma fórmula de desempenho foi desenvolvida para realizar uma análise comparativa da arquitetura do roteador com cada um dos decrementadores. O decrementador com transporte em cascata obteve a melhor relação de desempenho desde que seu tamanho esteja

entre 4 e 7 bits. A partir de 8 bits, três decrementadores que utilizam a técnica de transporte selecionado obtiveram desempenho semelhante, se diferenciando com relação à implementação do bloco de subtração. Dos três, o decrementador com transporte cascata-selecionado obteve o melhor desempenho.

Todo projeto que possuir um sistema digital composto por decrementadores sem o bit de transporte final poderá fazer uso das fórmulas aqui desenvolvidas para definir a estrutura que melhor se adequará às condições estabelecidas pelo projetista. Desta forma, será possível automatizar o processo de identificação do circuito digital mais apropriado a ser incorporado em um determinado sistema. Assim, pode-se reduzir custo e economizar tempo na realização de um projeto, bem como auxiliar os projetistas na obtenção de sistemas com melhores desempenhos.

Abstract. A Network-On-Chip is under development as a reconfigurable hardware platform onto which one can directly implement Petri Nets. To determine the best subtractor architecture to be used with the network routers a series of equations were developed to compute the number of logic gates and of logic levels both of the subtractor and the router. A formula has been established to comparatively assess the performance of each subtractor when used with the router architecture.

Keywords. Router, subtractors, performance.

Referências

- [1] K.M. Büyüksahin, F.N. Najm, High-level area estimation, em “ISLPED ’02: Proceedings of the 2002 international symposium on Low power electronics and design”, pp. 271-274, New York, NY, USA, 2002.
- [2] J.P. Deschamps, G.J.A. Bioul, G.D. Sutter, “Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems”, Wiley-Interscience, New Jersey, 2006.
- [3] D.S. Gelosh, D.E. Steliff, Modeling layout tools to derive forward estimates of area and delay at the rtl level, *ACM Trans. Des. Autom. Electron. Syst.*, **5**, No. 3 (2000), 451-491.
- [4] G.D. Micheli, L. Benini, “Networks on Chips: Technology and Tools (Systems on Silicon)”, Morgan Kaufmann Publishers, San Francisco, 2006.
- [5] M. Nemani, F.N. Najm, High-level area and power estimation for vlsi circuits, em “ICCAD ’97: Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design”, pp. 114-119, Washington, DC, USA, 1997.
- [6] C. Ramachandran, F.J. Kurdahi, D.D. Gajski, A.C.-H. Wu, V. Chaiyakul, Accurate layout area and delay modeling for system level design, em “ICCAD ’92: Proceedings of the 1992 IEEE/ACM International Conference on Computer-aided design”, pp. 355-361, Los Alamitos, CA, USA, 1992.
- [7] A. Srinivasan, G. Huber, D. LaPotin, Accurate area and delay estimation from rtl descriptions, *IEEE Transactions on VLSI Systems*, **6**, No. 1 (1998), 168-172.