

# Síntese de Contadores Síncronos de Alta Latência Baseados em Latches Transparentes

Duarte L. Oliveira<sup>1</sup>, Tiago Curtinhas<sup>1</sup>, Lester A. Faria<sup>1</sup>, Leonardo Romano<sup>2</sup>

<sup>1</sup>Divisão de Engenharia Eletrônica – Instituto Tecnológico de Aeronáutica – ITA – IEEA – SJC – SP – Brasil

<sup>2</sup>Departamento de Engenharia Elétrica do Centro Universitário da FEI – SBC – SP – Brasil

**Resumo** — A memória da parte sequencial de um sistema digital síncrono convencionalmente é composta por elementos de memória do tipo flip-flops. Esse tipo de elemento de memória além de ter uma alta potência dissipada também tem um tempo de atraso significativo (tempo de setup + tempo de propagação). Um componente sequencial muito importante em um sistema digital são os contadores síncronos. Contadores síncronos são convencionalmente sintetizados a partir de Flip-Flops. Neste artigo propomos um novo método para projeto de contadores síncronos que são sintetizados a partir de latches transparentes. A substituição Flip-Flops por latches transparentes acarretam algumas potenciais vantagens, como redução do tempo de atraso, que significa aumento de desempenho e redução no tempo setup, que aumenta a robustez à metaestabilidade. Através de dois casos de estudo mostramos a simplicidade e a eficiência do novo método, como também uma melhoria nos resultados.

**Palavras chaves** — síntese lógica, contadores, divisores de frequência, máquinas de estado finito.

## I. INTRODUÇÃO

A fase corrente da microeletrônica, que está em nanômetros (*Deep-Sub-Micron* – DSM-CMOS) permite o projeto VLSI (*Very Large Scale Integration*) de sistemas digitais síncronos de alta complexidade, e muitos desses projetos tem com obstáculos a distribuição do sinal de relógio (*clock*) e exigência de elevado desempenho [1,2]. Na tecnologia DSM-CMOS o atraso nas linhas podem ser maior que o atraso nas portas, como também aumenta a intolerância ao ruído e a interferência eletromagnética [3]. O sinal de *clock* é o principal causador tanto do ruído como interferência eletromagnética. Finalmente, podemos dizer que na tecnologia DSM-CMOS dois importantes desafios são: a distribuição do *clock* e resolver o problema de defasagem do *clock* (*clock skew*).

Muitos sistemas digitais síncronos contemporâneos são descritos por uma arquitetura que consiste de uma rede de controladores + *datapaths* (ASIC – *Application Specific Integrated Circuit*) e / ou processadores [4,5]. Os controladores de tais circuitos síncronos são muitas vezes referidos como uma máquina de estado finito (MEF) composta de vários estados e transições entre estados, ou seja, são especificados por um grafo de transição de estados (GTE).

Em um sistema digital a parte sequencial é a principal contribuinte para a dissipação de potência dinâmica. Estudos

recentes têm mostrado que em tais sistemas o *clock* consome uma grande percentagem (15% a 45%) da potência do sistema [6]. O sinal de *clock* atua diretamente nos elementos de memória, do tipo *flip-flop* ou *latch*. Os métodos de síntese de MEFs síncronas usam *flip-flops* (*single-edge-transition* – SET-FF) para armazenar os estados da máquina, mas este elemento de memória contribui para o problema de *clock skew* e seu tempo de atraso é significativo (tempo de *setup* + tempo de propagação) afetando o desempenho [7-9]. Uma alternativa promissora para a síntese de MEFs síncronas de baixo consumo de energia é substituir os *flip-flops* por *latches* transparentes. Os *latches* transparentes quando comparados com os *flip-flops* têm várias vantagens, tais como: a) o consumo de energia é até um terço menor que os *flip-flops* [10]; b) área é até um terço menor que os *flip-flops* [10] (ver Fig. 1); c) problema da defasagem do *clock* é reduzida; d) os tempos de *setup* e de propagação são menores; e) Além disso, a menor capacitância do *clock* no *latch* pode reduzir ainda mais o consumo de energia na árvore do *clock*. Uma vez que a maior parte do consumo de energia em recentes *chips* é devida à árvore do *clock* [11].

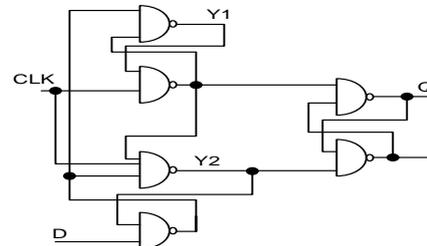


Fig. 1. SET D Flip-Flop: arquitetura clássica.

Diferentes abordagens têm sido propostas para sintetizar MEFs síncronas utilizando *latches* transparentes. Em [12,13] utiliza uma estratégia de *clock* de múltiplas fases. Em [14]-[17] usam *clock* de simples fase, mas diferentes estratégias para a substituição de alguns *flip-flops* por *latches*. Estas estratégias têm limitações; o uso de múltiplas fases aumenta a complexidade de distribuição do *clock* e a substituição parcial de *flip-flops*, conduz a uma otimização subótima do desempenho. Houve algumas abordagens para usar *latches* em vez de *flip-flops* para obter esses méritos [12],[18]-[20]. Em [12] mostrou uma economia de energia de até 50% quando os *latches* foram usados. Mas o uso de esquema de múltiplos *clock's* é muito desagradável para projetistas de VLSI. Em [18], elementos de armazenamento implementados com os *flip-flops* são substituídos com *latches* se o comportamento do circuito de entrada / saída não é afectada. Em [19] usa a técnica assíncrona para decidir quais *flip-flops* podem ser substituídos por *latches* e usa o esquema de *clock* de fase simples. Em [20] usa a técnica de mapeamento direto

no estilo *one-hot*, onde os latches operam nos dois níveis do clock, permitindo operar com a metade da frequência, mas tendo a mesma taxa de processamento.

Contadores síncronos são importantes componentes em um sistema digital. Eles são MEFs síncronas modelo Moore, que são descritas por um GTE. Os contadores síncronos usam *flip-flops* como elemento de memória, e uma forma muito simples de usar *latches* transparentes é substituir os *flip-flops* por dois *latches* transparentes operando no estilo mestre escravo. A operação do contador neste estilo permite enxergar o desbalanceamento que ocorre nos níveis alto e baixo do sinal de *clock*, ocorrendo uma ociosidade em um dos níveis.

Neste artigo propomos um novo método baseado no particionamento dos estados para síntese de contadores síncronos. O método proposto sintetiza os contadores síncronos usando como elemento de memória somente *latches* transparentes, sendo o sinal de *clock* de simples fase. O nosso projeto permite que os *latches* transparentes operem na forma balanceada, eliminando a ociosidade em uma das partes do sinal do *clock*. Os nossos contadores são implementados na arquitetura alvo da Fig. 2. Uma aplicação muito promissora do nosso método quando comparada com a síntese clássica é o projeto de divisores de frequência por um número ímpar gerando um sinal de saída com nível simétrico que é baseado em contadores. Através de um caso de estudo mostramos o projeto de um contador reversível com um melhor tempo de latência quando comparado com a síntese clássica. Também mostramos uma aplicação promissora dos nossos contadores, que é o divisor de frequência.

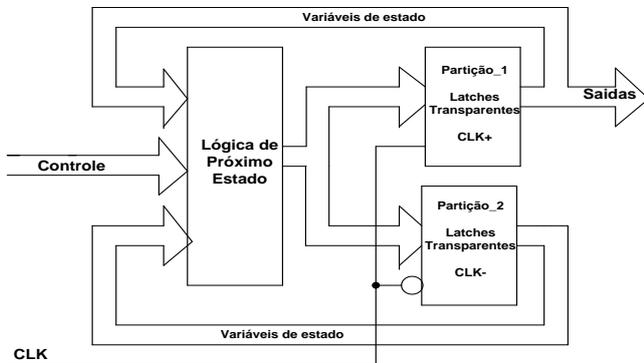


Fig. 2. Arquitetura alvo proposta: contador síncrono.

O restante deste artigo está organizado da seguinte forma. A seção II discute a síntese clássica dos contadores síncronos; a seção III mostra o procedimento proposto de particionamento e síntese de contadores síncronos; a seção IV apresenta um caso de estudo; a seção V mostra a aplicação do nosso método no projeto de divisores de frequência; seção VI mostra a simulação de dois casos de estudo e resultados; finalmente a seção VII apresenta as nossas conclusões e futuro trabalho.

## II. SÍNTESE DE CONTADORES: ANÁLISE

A síntese clássica de contadores usam *flip-flops* como elementos de memória. Os tempos de setup e de propagação dos *flip-flops* são bastante significativos, interferindo significativamente nos tempos de latência e de ciclo dos contadores. Figura 3 mostra o GTE de um contador síncrono

reversível módulo quatro. Figura 4 mostra o circuito lógico convencional deste contador, usando *flip-flop* do tipo D. Figura 5 mostra o mesmo contador, mas usando *latches* transparentes operando como mestre escravo. A substituição dos elementos de memória não reduz os tempos de latência e de ciclo. O tempo de latência está relacionado com o tempo de atraso que começa com a transição do sinal do *clock* até a ativação do sinal de saída. O tempo de ciclo está relacionado com o atraso do pior caminho entre dois elementos de memória. Começa no tempo de propagação do primeiro elemento de memória até o tempo de *setup* do segundo elemento de memória. Figura 6 mostra outra configuração, onde as saídas são obtidas no latch mestre. Nesse caso, há redução substancial no tempo de latência, mas quando há sinais de entrada, no caso o sinal *Dir*, surge outro problema, que acarreta um aumento drástico do tempo de *hold* e que é equivalente ao nível alto do sinal do *clock*.

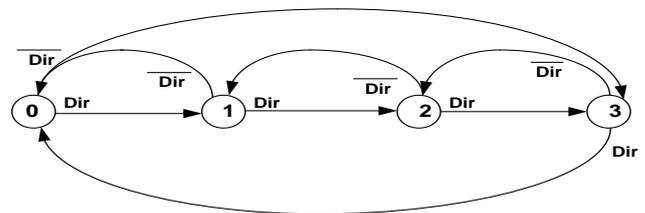


Fig. 3. Contador reversível módulo quatro: GTE.

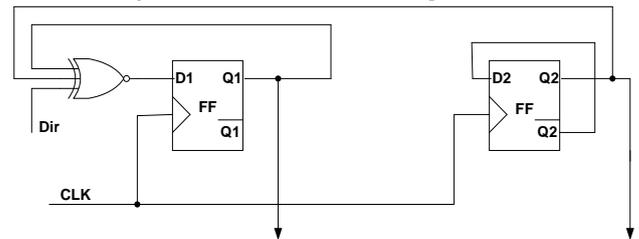


Fig. 4. Contador reversível módulo quatro: circuito lógico com SET D FF.

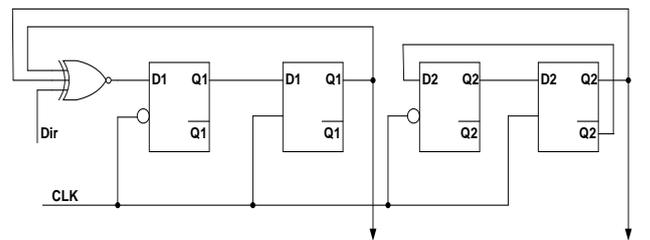


Fig. 5. Contador usando latches transparentes como mestre escravo.

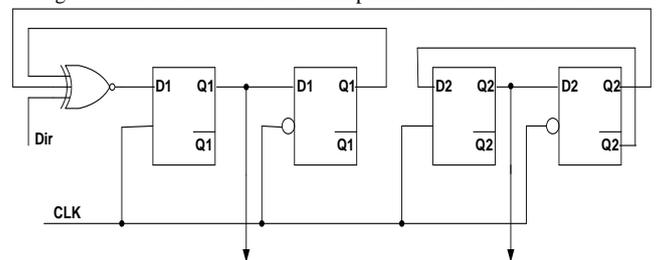


Fig. 6. Contador usando latches transparentes com tempo de hold.

## III. ESTILO ME\_PARTIÇÃO: CONTADORES

O método proposto neste artigo particiona os estados em duas partições ( $P_1, P_2$ ), que é visto como o particionamento da memória do contador (ME\_Partição). Na partição  $P_1$  os estados são armazenados em *latches* transparentes e ativados

no nível alto. Na outra partição  $P_2$  os estados são armazenados em latches transparentes ativados no nível baixo.

A partição  $P_1$  é formada pelo conjunto de estados do grafo de transição de estados (GTE). A partição  $P_2$  é formada pelo conjunto de estados NOP (*non-operation*) inseridos nas transições de estados do GTE, formando um novo GTE, denominado GTE\*. Cada partição está relacionada com um semi-código, portanto um estado do GTE\* é codificado com dois semi-códigos. Eles são chamados de positivo quando relacionado com o *clock* CLK+ e negativo quando relacionado com o *clock* CLK-. A concatenação (símbolo usado é &) dos dois semi-códigos formam o código do estado do GTE\*. Cada transição de estado do GTE\* somente um semi-código pode ser ativado.

O assinalamento de estados do GTE\* é realizado em duas etapas. Na primeira etapa o semi-código positivo binário dos estados da partição  $P_1$ , já estão codificados, para o caso dos contadores. A segunda etapa o semi-código negativo binário relacionado com os estados da partição  $P_2$  devem ser codificados, independentemente do outro semi-código.

O nosso algoritmo de particionamento/assinalamento para os nossos contadores segue três passos:

1. Capturar a descrição do contador síncrono no GTE.
2. Inserir estados NOP em todas as transições de estado do GTE, obtendo um novo GTE (GTE\*) e gerar as partições  $P_1$  e  $P_2$ .
3. Realizar o assinalamento de estados NOP no GTE\*, no caso a partição  $P_2$  e preferencialmente usando o código Gray.

Para ilustrar o nosso algoritmo, aplicamos no GTE de um contador reversível modulo oito da Fig. 7. Figura 8 mostra o correspondente GTE\* (passo 2). Os estados NOP foram inseridos em todas as transições de estado do GTE, com o símbolo de estados ativados em baixa. Figura 9 mostra o GTE\* com os semi-códigos binários. No semi-código positivo de  $P_1$  é 8 [0 até 7]. No semi-código negativo é 16 [0 até 15]. (passo 3). A partição  $P_1\_CLK+$  contém as variáveis de estado  $Q_1$ ,  $Q_2$  e  $Q_3$ , portanto os latches transparentes correspondentes operarão no nível alto do *clock*. A partição  $P_2\_CLK-$  contém as variáveis de estado  $Q_4$ ,  $Q_5$ ,  $Q_6$  e  $Q_7$ , portanto os latches transparentes correspondentes operarão no nível baixo do *clock*. A codificação binária do semi-código negativo obedece á distância mínima de Hamming, portanto é uma codificação de baixa potência.

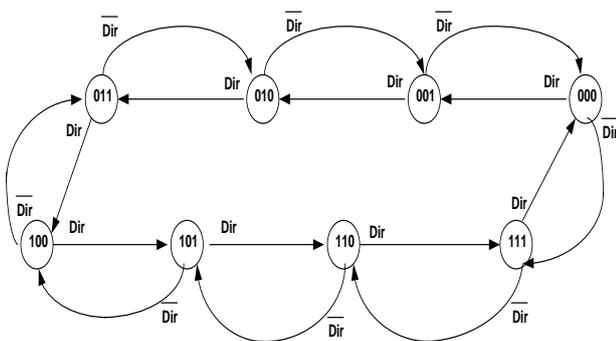


Fig. 7. GTE: contador reversível módulo oito.

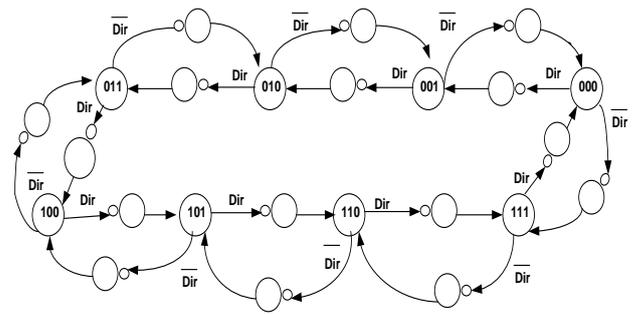


Fig. 8. GTE\*: contador reversível módulo oito com estados NOP.

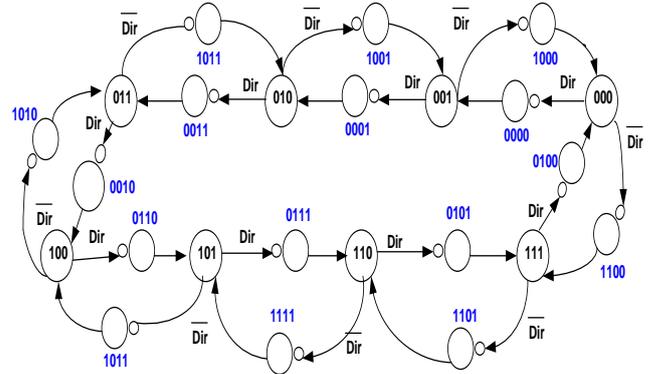


Fig. 9. GTE\*: contador reversível módulo oito com estados NOP codificado.

#### A. Método proposto: contadores

A nossa metodologia implementa os contadores na arquitetura mostrada na Fig. 1. Ela é composta de quatro passos:

1. Gerar GTE\* codificado (seção III)
2. Montar a tabela verdade onde as entradas são as variáveis de estado da partição  $P_2$  e as variáveis de saída são as variáveis e estado da partição  $P_1$ . Extrair a para cada variável de saída a equação Booleana minimizada de próximo estado na forma de soma de produto.
3. Montar a tabela verdade onde as entradas são as variáveis de estado da partição  $P_1$  e as variáveis de saída são as variáveis e estado da partição  $P_2$ . Extrair a para cada variável de saída a equação Booleana minimizada de próximo estado na forma soma de produto.
4. Montar a tabela verdade onde as entradas são variáveis de  $P_1 \cup P_2$  e as saídas se houver são os sinais de saída. Extrair a para cada variável de saída a equação Booleana minimizada soma de produto.

### IV. CASO DE ESTUDO: CONTADOR REVERSÍVEL

Nesta seção, ilustramos o nosso método de síntese com o contador síncrono reversível módulo quatro (ver Fig. 3). O primeiro passo aplica o algoritmo da seção III e gera o GTE\* codificado como mostra as Fig. 10 e 11. A codificação binária resultante necessitou na partição  $P_1\_CLK+$  das variáveis  $Q_1$  e  $Q_2$  e na partição  $P_2\_CLK-$  das variáveis  $Q_3$ ,  $Q_4$  e  $Q_5$  (ver Fig. 9). Fig. 12 mostra a tabela de transição de estados do contador reversível. Fig. 13 mostra a minimização lógica da partição  $P_2$  e Fig. 14 mostra a minimização lógica da partição  $P_1$ . Fig. 15 mostra o circuito lógico do contador reversível.

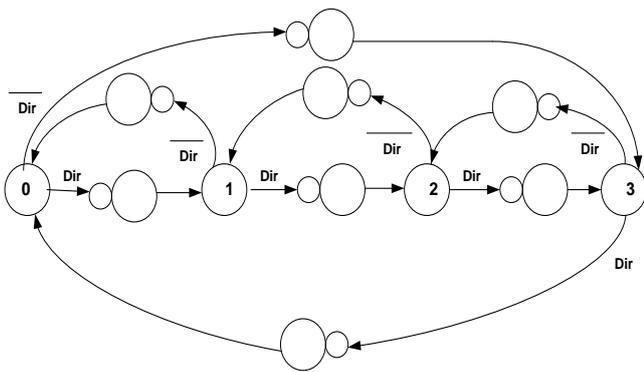


Fig. 10. GTE\*: contador reversível módulo quatro com estados NOP.

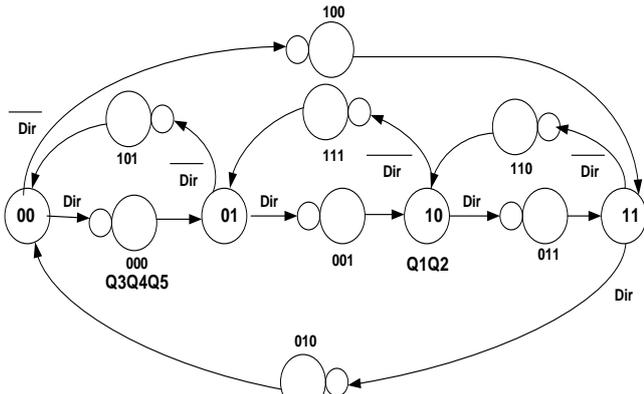


Fig. 11. GTE\* codificado: P<sub>1</sub>\_CLK+ (Q1,Q2) e P<sub>2</sub>\_CLK- (Q3,Q4,Q5).

Estado atual Q2 Q1	Próximo estado	
	Dir=0 Q5Q4Q3	Dir=1 Q5Q4Q3
0 0	1 0 0	0 0 0
0 1	1 0 1	0 0 1
1 1	1 1 0	0 1 0
1 0	1 1 1	0 1 1

Fig. 12. Tabela de transição de estados.

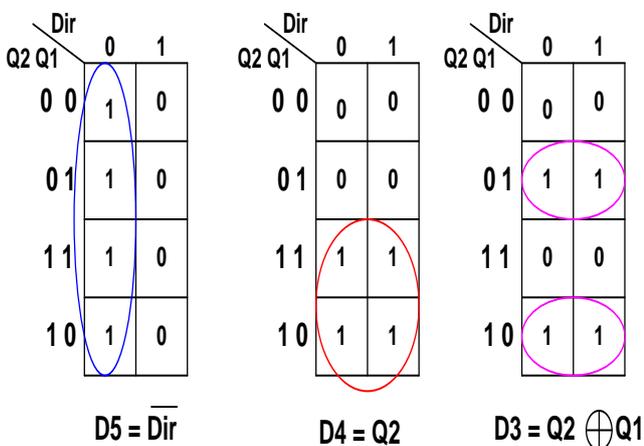
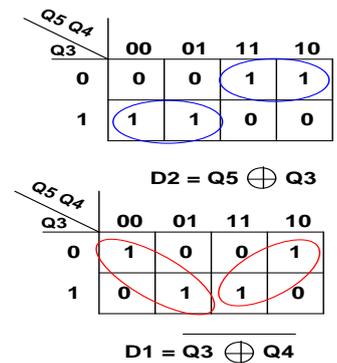


Fig. 13. Map de Karnaugh: solução dos estados NOP.

Estado atual Q5 Q4Q3	Próximo estado Q2 Q1	
	Dir=0	Dir=1
0 0 0	0 1	0 0
0 0 1	1 0	0 0
0 1 0	0 0	1 1
0 1 1	1 1	0 0
1 0 0	1 1	1 0
1 0 1	0 0	0 1
1 1 0	1 0	0 1
1 1 1	0 1	0 0

(a)



(b)

Fig. 14. Estados do contador: a) Tabela de transição de estados; b) Mapa de Karnaugh.

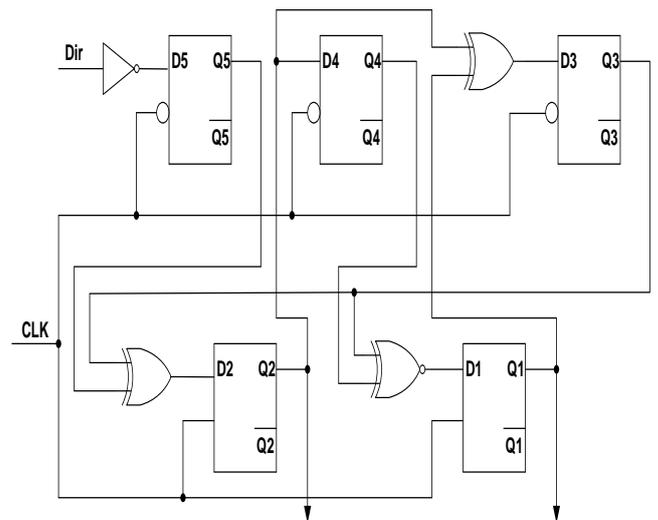


Fig. 15. Circuito lógico do contador reversível: solução baseada em latches.

### V. APLICAÇÃO: DIVISOR DE FREQUÊNCIA

Divisores de frequência são importantes no projeto de sintetizadores de frequência. Comunicações sem fio tem recebido um crescente interesse nas aplicações militares e comerciais. As demandas por comutação rápida e frequências de operação elevadas fazem o projeto de sintetizadores de frequência uma tarefa difícil [21-23]. Sintetizadores de frequência pode ser de um tipo inteiro-N com divisores de frequência inteiros programável ou do tipo fracional.

Para ilustrar o nosso método, aplicamos no projeto de um divisor de frequência por cinco com saída simétrica (*duty cycle* 50%). Figura 16 mostra o diagrama de temporização do divisor de frequência desejado. Figura 17 mostra o projeto convencional de um divisor de frequência por cinco com saída simétrica [22]. Para reduzir a lógica de saída, deve-se encontrar uma sequência que mais se enquadra no problema. Como o DF é por cinco, precisamos de um gerador de sequência módulo cinco. No nosso método a frequência de saída é formada por dois geradores de sequência, portanto a Fig. 18a mostra uma geração de sequência interessante, para a parte que vai ser ativada no nível alto. A variável Q1 se comporta como um divisor por cinco, mas com 2/3 de *duty cycle*. Devemos gerar uma sequência para a parte que vai ser ativada no nível baixo, tal que uma variável possa ajustar o *duty cycle*. Fig. 18b mostra uma sequência ativada no nível

baixo, onde a variável  $Q4$  realiza o ajuste. Fig. 18c mostra a introdução do sinal  $F_{sai}$  no GTE. Usando o método proposto a Fig. 19 mostra o circuito lógico do divisor por cinco.

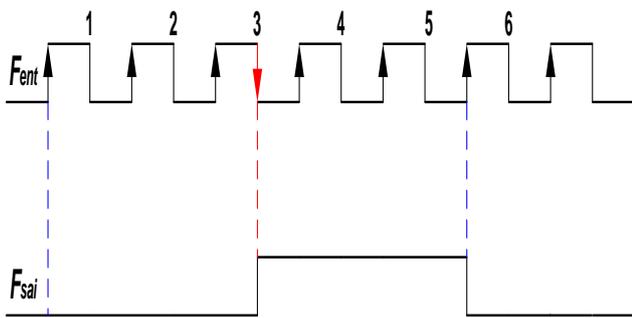


Fig. 16. Diagrama de temporização: divisor por cinco de frequência simétrica.

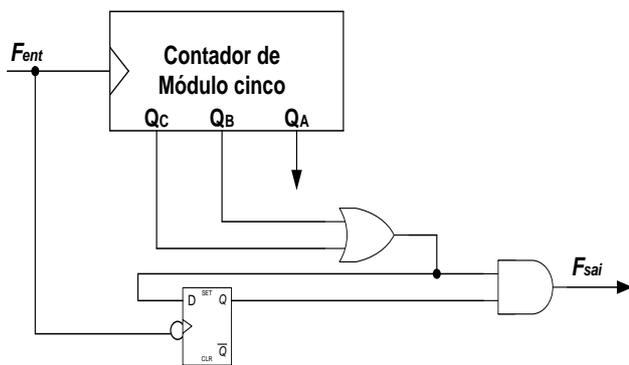


Fig. 17. Circuito lógico: divisor/5 de F simétrico - solução clássica.

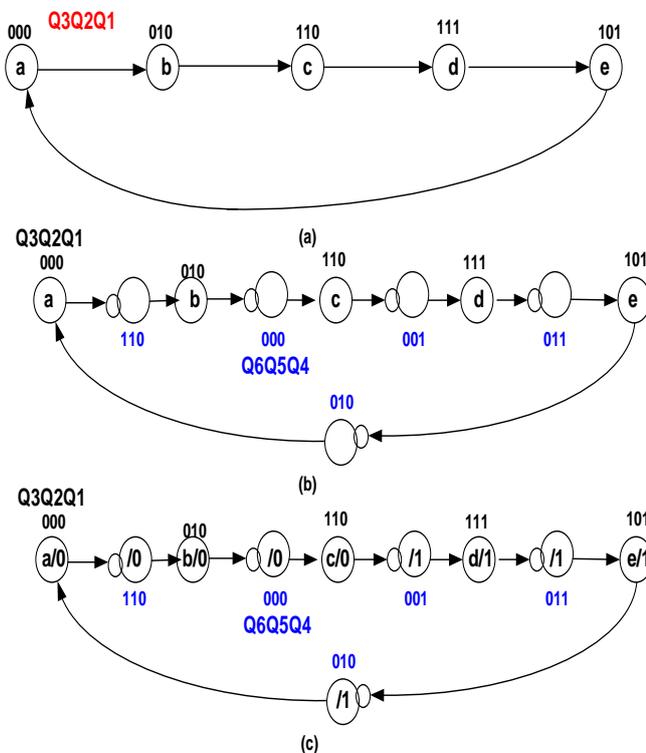


Fig. 18. GTE: a) gerador de sequência módulo cinco; b) gerador de sequência módulo cinco com estados NOP; c) divisor de frequência por cinco com frequência simétrica.

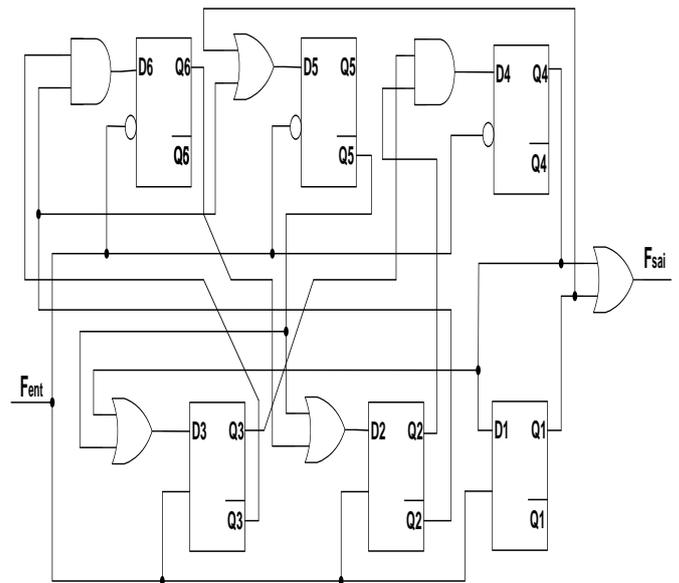


Fig. 19. Divisor de F/5 simétrico: circuito lógico baseado em latches.

## VI. DISCUSSÃO & SIMULAÇÃO & RESULTADOS

Contadores síncronos são importantes em um sistema digital. Como são bastante acionados, tem uma tendência de alto consumo de energia. Nas bibliotecas *standard cell* convencionais os dados de área e desempenho para *latches* transparentes são reduzidos, quando comparado com *flip-flops*. A substituição de *flip-flops* por *latches* traz diversas vantagens no projeto, tais como: a) redução do tempo de *setup*, o que torna o circuito mais robusto a metaestabilidade [24]; b) aumento da tolerância a *clock skew*; c) redução de área e potência dissipada.

As Figuras 20,21, e 22 mostram a simulação das três diferentes técnicas de projeto de contadores, apresentando os respectivos tempos de latência. As simulações foram realizadas no software da ALTERA QUARTUS II, versão 9.1, na família CYCLONE III, dispositivo EP3C16F484C6N [25,26].

Tabela I traz a análise de área (LUTs + Flip-Flops) e tempo de latência de quatro diferentes implementações no dispositivo FPGA do contador reversível de módulo quatro. A técnica proposta obteve uma redução máxima no tempo de latência de 55% e uma penalidade máxima em área de 225%. no dispositivo FPGA. A potência dissipada obtida nas quatro técnicas praticamente é equivalente.

Fazendo uma comparação em VLSI no quesito de área com a síntese convencional e usar como referência os FFs de Fig. 1, onde SET-FFs D são sintetizados com 26 transistores e os *latches* transparentes com 18 transistores. O nosso método precisou de 110 transistores para o contador reversível, enquanto que o método convencional precisou de 70 transistores, portanto houve uma penalidade de 57%. O projeto de contadores síncronos em VLSI baseados em *latches* transparentes são promissores para obter alto desempenho no tempo de latência [27].

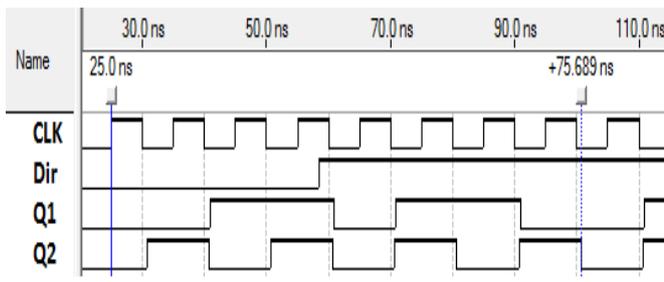


Fig. 20. Simulação: contador reversível módulo quatro.

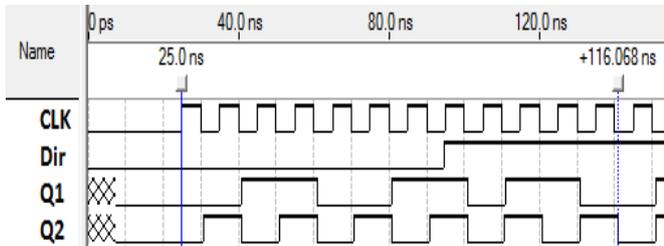


Fig. 21. Simulação: contador reversível módulo quatro baseado em mestre escravo.

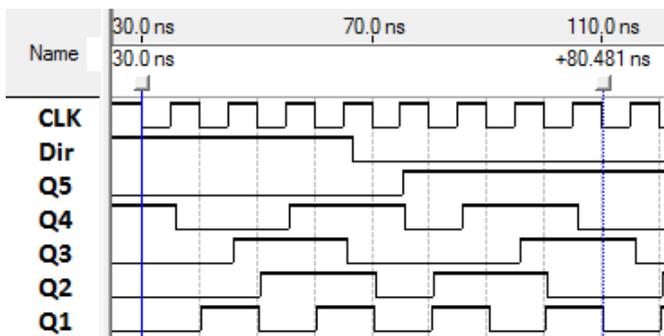


Fig. 22. Simulação: contador reversível módulo quatro baseado em latches transparentes.

TABELA I RESULTADOS

Figuras	Tempo de Latência	Número de LUTs	Número de FFs	Dissipação de Potência
4	1546ps	2	2	1.33mw
5	956ps	9	0	1.34mw
6	912ps	9	0	1.35mw
<b>15</b>	<b>688ps</b>	13	0	1.35mw

## VII. CONCLUSÃO

Neste artigo apresentamos um método para síntese de contadores síncronos baseados em *latches* transparentes. O novo método também permite sintetizar divisores de frequência de um número ímpar com frequência simétrica, numa forma elegante e eficiente. Os divisores de frequência sintetizados pelo novo método apresentam um ótimo tempo de latência, além de uma área reduzida. Para muitos tipos de contadores, este estilo consegue uma redução de área, quando comparado com a síntese convencional. Os contadores síncronos também podem ser usados na implementação de qualquer MEF síncrona. Como trabalho futuro pretendemos aplicar o estilo ME-partição na síntese de processadores programáveis. A grande vantagem do método proposto está

na temporização, onde o tempo de hold é zero e o tempo de setup e reduzido.

## REFERÊNCIAS

- Unger, S., "Clocking Schemas for High-Speed Digital Systems," *IEEE Trans. on Computers*, vol. c-35, no. 10, October, pp.880-895, 1986.
- D. Goldhaber-Gordon, et al., "Overview of Nanoelectronic Devices," *Proc. of the IEEE*, vol. 85, No. 4, pp.521-540, April 1997.
- B. H. Calhoun, et al. "Digital Circuit Design Challenges and Opportunities in the Era of Nanoscale CMOS," *Proceedings of the IEEE*, Volume 96, Issue 2, February 2008.
- H. Hsieh, F. Balarin et al. "Synchronous approach to the Functional Equivalence of Embedded System Implementations," *IEEE Trans. on CAD of Int. Circuits and Systems*, vol.20, no.8, pp.1016-1033, August 2001.
- L. Jozwiak, et al., "Multi-objective Optimal Controller Synthesis for Heterogeneous embedded Systems," *Int. Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation*, pp. 177-184, 2006.
- A. Jain et al., "A 1.2 GHz alpha microprocessor with 44.8 GB/s chip pin bandwidth," in *IEEE Int. Solid-State Circuits Conf. Tech. Dig.*, pp. 240-241, Feb. 2001.
- G. M. Strollo et al., "Power Dissipation in One-Latch and Two-Latch Double Edge Triggered Flip-Flops," *Proc. 6th IEEE Int. Conf. on Electronic, Circuits and Systems*, pp.1419-1422, 1999.
- S. H. Rasouli, A. Kahademzadeh and et al. "Low-power single- and double-edge-triggered flip-flops for high-speed applications," *IEE Proc. Circuits Devices Syst.*, vol. 152, no. 2, pp.118-122, April 2005.
- P. Zhao, J. McNeely, et al., "Low-Power Clock Branch Sharing Double-Edge Triggered Flip-Flops," *IEEE Trans. On VLSI Systems*, vol. 15, no.3, pp.338-345, March 2007.
- 0.65um Cell Based Library Databook, LG Semicon, 1995.
- M. Pedram and A. Abdollahi, "Low-Power RT-Level Synthesis Techniques: a Tutorial," *IEE Proc. Comput. Digit. Tech.*, vol. 152, no. 3, pp. 333-343, May 2005.
- Bartlett, K. and Borriello, G., "Timing Optimization of Multiphase Sequential Logic," *IEEE Trans. on CAD*, vol.10, no. 1 January, pp.51-62, 1991.
- Christos A. Papachristou, Mehrdad Nourani, and Mark Spining. "A Multiple Clocking Scheme for Low-Power RTL Design," *IEEE TVLSI*, vol. 7, no. 2, June 1999.
- C. Leiserson, F. Rose, and J. Saxe, "Optimizing Synchronous Circuits by Retiming," *Third Caltech Conference on VLSI*, pp. 87-116, 1983.
- Tsung-Yi Wu, Youn-Long Lin, "Storage Optimization by Replacing Some Flip-Flops with Latches," *Proc. EDAC*, 1996.
- Maheshwari, N., "In Fast algorithms for retiming large digital circuits," PhD Thesis, Iowa State University, 1998.
- S. Chattopadhyay, et al. "State Assignment and Selection of Types and Polarities of Flip-Flops, for Finite State Machine Synthesis," *IEEE India Conf. (INDICON)*, pp.27-30, 2004.
- W. Yang, et al., "Low-Power High-Level Synthesis using Latches," *IEEE Proc. of the ASP-DAC*, pp.462-465, 2001.
- D. L. Oliveira, et al., "Synthesis of High-Performance Synchronous Finite State Machine using Transparent Latches," *XV Iberchip*, 2009.
- D. L. Oliveira et al., "An Novel Approach for Design of Synchronous Finite State Machines using Transparent Latches," *XX Workshop Iberchip*, Santiago do Chile, 2014.
- R. Mohamavelu and P. Heydari, "A Novel Ultra High-Speed Flip-Flop-Based Frequency Divider," *Proc. IEEE ISCAS*, pp. IV 169-172, 2004.
- J. Zhao, "The Applications of MSI Counter—74X161," *Proc. IEEE* pp.1326-1329, 2011.
- G. Bandhawarkar Wakhle, "Synthesis and Implementation of UART using VHDL Codes," *Proc. IEEE International Symposium on Computer, Consumer and Control*, 2012.
- S. H. Unger, "Hazards, critical races, and metastability," *IEEE Transaction on Computer*, vol. 44, Issue:6, pp.754-768, June 1995.
- J. J. Rodriguez, et. al., "Features, Design Tools, and Applications Domains of FPGAs," *IEEE Trans. On Industrial Electronics*, vol. 54, no 4, pp. 1810-1823, August, 2007.
- Internet: [www.altera.com](http://www.altera.com), 2015.
- G. DE Micheli, "An Outlook on Design Technologies for Future Integrated Systems," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 28, no.6, pp. 777-789, June 2009.