

# Síntese Automática de Controladores Síncronos com Baixa Atividade do Relógio

\*Jozias Del Rios<sup>1</sup>, Duarte L. Oliveira<sup>1</sup>, Leonardo Romano<sup>1,2</sup>

<sup>1</sup>Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – ITA – IEEA  
Marechal Eduardo Gomes, 50 – CEP 12.228-900 – SJC – SP – Brazil  
email: [joziasdelrios@gmail.com](mailto:joziasdelrios@gmail.com), [duarte@ita.br](mailto:duarte@ita.br)

<sup>2</sup>Departamento de Engenharia Elétrica do Centro Universitário da FEI  
São Bernardo do Campo – São Paulo – Brasil  
email: [leoroma@uol.com.br](mailto:leoroma@uol.com.br)

**Resumo** — Em um sistema digital a atividade do sinal de relógio é um dos grandes consumidores de energia. Ele consome de 15% a 45% da energia consumida. Neste artigo nós propomos um método para síntese automática de controladores síncronos com baixa atividade no sinal de relógio. A redução da atividade do relógio é obtida através da aplicação de duas estratégias. Na primeira estratégia os nossos controladores operam nas transições de ambas as bordas do sinal de relógio, permitindo uma redução de 50% na frequência do sinal de relógio, mas com o mesmo tempo de processamento. Uma característica importante é que os nossos controladores usam somente Flip-Flops que são sensíveis a uma simples borda do sinal de relógio (*single edge transition – SET-FF*). Na segunda estratégia o sinal de relógio é inibido nos nossos controladores quando encontra um estado com auto-transição (*self-loop*).

**Palavras chaves** — Síntese lógica, baixo consumo, SET-Flip-Flop, máquinas de estado finito, controle de inibição de relógio.

## I. INTRODUÇÃO

Com a evolução da microeletrônica, cada vez mais são concebidos sistemas digitais de alta complexidade. Uma característica comum em uma parte destes sistemas é o fato de serem alimentados por bateria, e são voltados para diferentes aplicações tais como: comunicação sem fio, computadores portáteis, aplicações aeroespaciais (satélites, mísseis, aviões, etc), automóveis, aplicações médicas, etc. Uma vez que eles são alimentados por bateria é desejável que as baterias tenham uma longa vida útil, portanto a potência dissipada é um parâmetro muito importante na concepção de tais sistemas [1]. Estes sistemas podem ser implementados na tecnologia VLSI e / ou FPGAs (*Field Programmable Gate Array*). Os dispositivos FPGAs tornaram-se um meio popular de implementar circuitos digitais. Tecnologia FPGA tem crescido consideravelmente nos últimos anos, gerando FPGAs com cerca de 50 milhões de portas, permitindo assim que sistemas digitais complexos possam ser programados em tais dispositivos [2].

Tradicionalmente, circuitos digitais são implementados com componentes construídos com tecnologia CMOS. A potência dissipada nos componentes CMOS segue a seguinte expressão [3]:

$$P_{TM} = I/2 \cdot C \cdot V_{DD}^2 \cdot f \cdot N + Q_{sc} \cdot V_{DD} \cdot f \cdot N + I_{leakage} \cdot V_{DD} \quad (1)$$

\*Jozias Del Rios é aspirante da aeronáutica e aluno do 5º ano de engenharia eletrônica.

onde:  $P_{TM}$  é a potência dissipada total média,  $V_{DD}$  é a tensão de alimentação,  $f$  é a frequência de operação,  $N$  é o fator de atividade de comutação, isto é, o número de transições na saída de uma porta ou de um Flip-Flop, e o fator  $Q_{sc}$  e  $C$  são respectivamente a quantidade de carga e a capacitância [3].

Na equação (1), o primeiro termo representa a potência dissipada dinâmica. O segundo termo representa a potência dissipada relacionada a corrente de curto. O terceiro termo representa a potência dissipada estática que está relacionada com a corrente de fuga. Na tecnologia CMOS estática a maior fração da potência dissipada ocorre durante a mudança de eventos (potência dinâmica) [3]. Potência dissipada média na porta  $g$  pode ser simplificada para o primeiro termo de (1):

$$P_{MÉDIA-g} = I/2 \cdot C_g \cdot V_{DD}^2 \cdot f \cdot N_{MÉDIA-g} \quad (2)$$

A síntese de máquinas de estado finito (MEF) desempenha um papel importante na concepção de circuitos digitais alimentados por bateria. Muitos circuitos digitais síncronos são descritos por uma arquitetura que consiste de uma rede de controladores + via de processamento de dados (*datapaths*) e / ou processadores [4,5]. Os controladores de tais circuitos síncronos são muitas vezes referidos como uma MEF composta de vários estados e transições entre estados, ou seja, são especificados por um grafo de transição de estados (GTE).

As técnicas de redução de potência dinâmica são aplicadas nos diferentes níveis do projeto digital [1,6]. Na síntese de controladores síncronos, propostas para reduzir a potência estão sendo oferecidas no nível lógico: **1)** controle lógico do relógio (*gated-clock*) [7,8]; **2)** Flip-flops sensíveis a ambas as bordas do sinal de relógio [9,10,11]; **3)** decomposição [12,13]; **4)** assinalamento de estados [14,15]; **5)** minimização lógica [16,17].

Em um sistema digital a parte seqüencial é a principal contribuinte para a dissipação de potência dinâmica. Estudos recentes têm mostrado que em tais sistemas o relógio consome uma grande porcentagem (15% a 45%) da potência do sistema [7]. Então, a potência do circuito pode ser consideravelmente reduzida se a atividade do relógio for reduzida. Entre as propostas para a redução da potência dinâmica nos controladores síncronos, os dois primeiros são muito interessantes.

A. Redução na atividade do relógio

A **primeira estratégia** (*gated-clock*) utiliza uma lógica adicional para inibir (parar) o sinal de relógio nos estados com auto-transição (*self-loop*). Para algumas MEFs a maioria dos ciclos relógio são utilizados em situações de auto-transição. Nestes estados há também potência dissipada dinâmica, porque os Flip-Flops (FFs) internamente mudam de estado, apesar de não ocorrerem alterações nas saídas. Benini et al [7] propôs um método e uma arquitetura alvo para controladores síncronos com inibição de relógio. Estes controladores operam na simples borda da transição do sinal de relógio. A **segunda estratégia** é utilizar nos controladores, FFs que são sensíveis nas bordas de subida / descida do sinal de relógio (*double-edge triggered – DET*) [11]. Para uma taxa de processamento de dados (*data throughput*), o DET-FFs requer 50% da taxa de relógio, conseqüentemente reduz a atividade do relógio. Se compararmos DET-FFs e os FFs operando na simples borda da transição do sinal de relógio (*single-edge triggered – SET*), vamos notar um aumento no consumo de energia e área (número de transistores) [10]. Uma abordagem promissora que reduz a atividade do sinal de relógio é a união das duas estratégias: lógica de controle de relógio + MEF que usam DET-FF. Esta abordagem traz dois problemas: **1)** maioria das bibliotecas VLSI *standard-cell* não incluem este tipo de FF (DET), e as macro-células das FPGAs usam SET-FF D [18]; **2)** para arquitetura proposta em [7], a inibição do sinal de relógio nos estados com auto-transição é sub-ótima quando aplicada em MEF Moore que utilizam DET-FFs.

Neste artigo propomos uma ferramenta para síntese automática de controladores síncronos modelo Moore. O nosso método reduz drasticamente a atividade do sinal de relógio. Esta redução é obtida através de duas estratégias. Na **primeira estratégia**, o nosso método sintetiza controladores síncronos que operam em ambas as bordas da transição do sinal de relógio, mas usa somente SET-FF-Ds. Na **segunda estratégia**, propomos uma nova arquitetura que inibe o sinal de relógio em ambas as bordas da transição do sinal de relógio em estados com auto-transição. Figura 1 mostra a arquitetura alvo modelo Moore usada para implementar os nossos controladores síncronos.

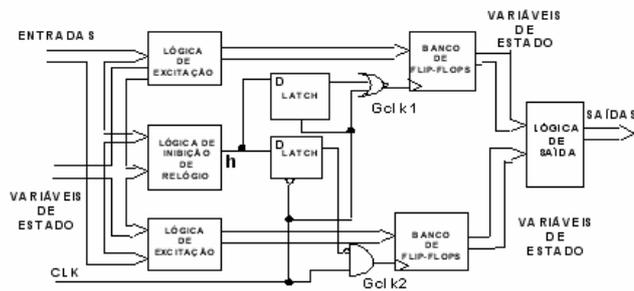


Fig. 1. Arquitetura alvo proposta: modelo Moore.

O restante deste artigo está organizado da seguinte forma. Na seção 2 apresentamos alguns conceitos para entendimento do nosso método; na seção 3 introduzimos o nosso método; na seção 4 ilustramos o nosso método com um exemplo da literatura; na seção 5 discutimos as vantagens e limitações do nosso método e alguns resultados; finalmente, na seção 6 apresentamos as nossas conclusões e futuro trabalho.

II. PRELIMINARES

Controlador síncrono é uma máquina de estados finito determinística modelo Moore ou Mealy cujo comportamento é descrito por um grafo de transição de estados (GTE). Os vértices representam estados e as arestas transições de estados. Figura 2 ilustra um GTE modelo Moore de um controlador de semáforo. O exemplo tem as entradas [T,Sr1,Sr2] e as saídas [A1,A2,V1,V2,v1,v2]. A idéia principal do nosso método é particionar o GTE modelo Moore em dois sub-conjuntos de transições de estado, tal que cada sub-conjunto está relacionado respectivamente com um banco de FF's. Um banco de SET-FF's operará na borda de subida (CLK+) do sinal de relógio e o outro banco de SET-FF's na borda de descida (CLK-). O particionamento é obtido através da construção de um grafo denominado de grafo de transição de relógio (GTR) proposto em [19].

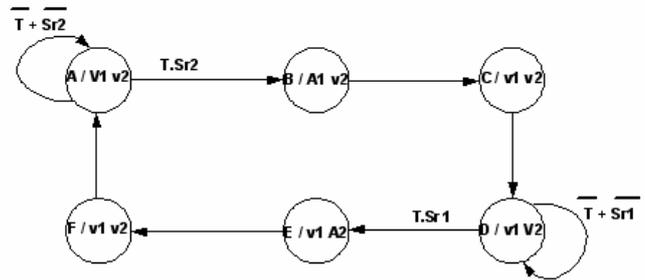


Fig. 2 Especificação: GTE modelo Moore.

A. Grafo de transição de relógio

Nesta seção apresentamos o GTR e conceitos para a sua manipulação. No GTR as transições de estado entre dois estados quaisquer do GTE é definida como **ponte**, porque a transição é não direcionada.

**Definição 2.1.** *Grafo de Transição de Relógio – GTR* é um grafo não direcionado  $\langle V, A, Est \rangle$ , onde  $V$  é o conjunto de vértices que descrevem estados,  $A$  é o conjunto de arestas que descrevem pontes e que são rotuladas com sinal de relógio que são polarizadas em  $\{+, -\}$ . O estado  $Est$  é o estado inicial.

Figuras 3a,b mostram respectivamente o GTE com entradas e saídas omitidas e o seu GTR sem rotulo (sinal de relógio). Figura 3a mostra seis transições de estado. Figura 3b mostra quatro pontes. A auto-transição no estado B não é uma ponte e as duas transições de estado entre os estados C e D formam uma única ponte (C-D). A interligação entre um conjunto de pontes forma um **caminho**. Se a interligação entre pontes inicia e termina em um mesmo estado define um **ciclo**. Figura 3b mostra o ciclo  $\{(A-B), (B-C), (C-D), (D-A)\}$ . Uma ponte será definida como **positiva** se ela estiver associada á borda de subida do sinal de relógio (CLK+). Se ela estiver associada á borda de descida do sinal de relógio será definida como **negativa** (CLK-). Um ciclo é chamado **degenerado** se todas as pontes pertencentes ao mesmo ciclo são do mesmo tipo, no caso positiva ou negativa. Um ciclo é par se ele possui um número par de pontes, caso contrário o ciclo é ímpar. O GTR é chamado degenerado se todas as pontes são do mesmo tipo. A MEFs é degenerada se o seu GTR é degenerado. Figuras 4a,b mostram respectivamente um ciclo par degenerado e o outro um ciclo par não degenerado. Figura 4a o GTR é degenerado.

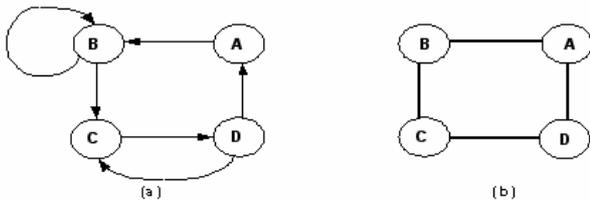


Fig. 3. Especificação: a) GTE omitindo entradas e saídas; b) GTR omitindo o sinal de relógio.

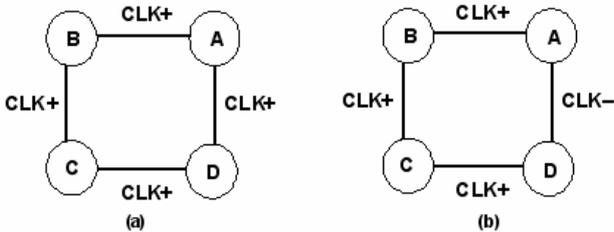


Fig. 4. GTR: a) ciclo par degenerado; b) ciclo par não degenerado.

III. SÍNTESE AUTOMÁTICA: MÉTODO

O nosso controlador é uma MEF modelo Moore especificada incompletamente. A nossa ferramenta sintetiza o nosso controlador usando flip-flops D. Ela segue o procedimento tradicional e possui cinco passos:

1. Capturar a descrição do controlador em um grafo de transição de estados modelo Moore (GTE). A ferramenta aceita a descrição do GTE ou no formato KISS2 [20], ou no formato proposto denominado EMS (*Explicit Machine Specification*) [21].
2. Realizar a minimização de estados no GTE usando o algoritmo de particionamento e obter o  $GTE_{MIN}$  [22]. Neste passo o algoritmo de particionamento que é voltado para especificações completas é modificado para aceitar especificações incompletas. A modificação usa uma heurística para especificar as saídas e estados don't-care.
3. A partir do  $GTE_{MIN}$  construir o grafo de transição de relógio (GTR) (seção A).
4. Codificar o  $GTE_{MIN}$  com redução de chaveamento (obter o  $GTE_{MIN-COD}$ ) (seção B)
5. A partir do  $GTE_{MIN-COD}$  usando o algoritmo expresso [20]<sup>2</sup>, extrair as equações minimizadas na forma soma de produto de excitação, saída e de inibição.

A. Geração do GTR

O GTR é obtido a partir do  $GTE_{MIN}$ . O nosso algoritmo é composto de três passos [21]:

1. Gerar o GTR e extrair o conjunto de pontes e de ciclos.
2. Ajustar a paridade de cada ciclo.
3. Definir rotulo (tipo) para as pontes.

No primeiro passo o algoritmo extrai as pontes ordenadas sem rotulo e todos os ciclos do GTR. Esta tarefa é realiza por percorrer por profundidade o  $GTE_{MIN}$ . O segundo passo verifica em cada ciclo a sua paridade. Para ciclos com um número ímpar de pontes, o nosso método opcionalmente

ajusta estes ciclos para um número par de pontes. O ajuste é realizado através da introdução de estados NOP. A vantagem é o particionamento equilibrado que permite uma maior redução na atividade do sinal de relógio e no número de variáveis de estados. Figura 5a mostra um GTR com um ciclo ímpar de pontes. Figura 5b mostra o GTR com ajuste de paridade através da introdução no estado NOP. O terceiro passo do algoritmo define (rotula) o tipo de cada ponte. Se o tipo for ponte negativa o rotulo é CLK-. Se o tipo for ponte positiva o rotulo é CLK+. O algoritmo procura definir no GTR uma alternância no tipo de ponte e a definição do tipo de ponte em um ciclo deve satisfazer o lema 3.1. Figura 4a não satisfaz o lema 3.1, porque somente há um único caminho de ponte positiva e negativa. Figuras 6a,b mostram respectivamente os ciclos desequilibrados e equilibrados. Eles satisfazem o lema 3.1 porque há pelo menos dois caminhos de pontes positivas e negativas. A introdução de estados NOP em um GTR é uma alternativa para satisfazer o lema 3.1.

**Lema 3.1** (sem Prova). Seja  $E=\{A,B,C,\dots,Z\}$  o conjunto de estados do GTR e  $Cy$  um ciclo qualquer do GTR, onde  $Cy=\{(AB),(BC),\dots,(XA)\}$ , sendo que  $(DK) \in Cy$  é uma ponte do GTR. Dizemos que o GTR não é codificável na forma não degenerada se e somente se em  $Cy$  houver um único caminho de pontes positiva ou negativa.

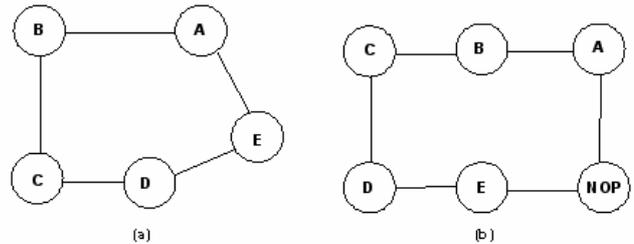


Fig. 5. GTR: a) ciclo ímpar de pontes; b) ajuste de ciclo.

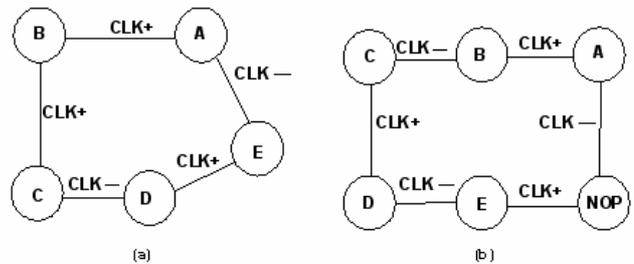


Fig. 6. GTR: a) ciclo desequilibrado; b) ciclo equilibrado.

B. Codificação de estados

Nesta seção descrevemos o algoritmo proposto para codificar o GTR. A codificação é realizada em três passos [21]:

1. Codificação simbólica
2. Redução de código
3. Codificação binária

B1. Codificação simbólica

A arquitetura alvo mostrada em Fig. 1 tem como característica o particionamento dos FF's em dois bancos. Um banco os FF's operam na borda de subida do sinal de relógio (B\_CLK+). O outro banco na borda de descida. (B\_CLK-). No primeiro passo o algoritmo proposto codifica

<sup>2</sup> A ferramenta de minimização lógica expresso da universidade de Berkley foi incorporada na nossa ferramenta.

simbolicamente o GTR. Cada estado do GTR é simbolicamente codificado e cada código simbólico é formado por dois semi-códigos simbólicos. Ele deve satisfazer a regra de partição de código (*regra\_pc*) e cada semi-código (valor inteiro) está relacionado com um banco de FF's. Esta codificação deve satisfazer a *regra\_pc* abaixo, onde cada estado é codificado simbolicamente com dois semi-códigos concatenados. Figuras 7a,b mostram o GTR codificado simbolicamente, onde em Fig. 7a a codificação satisfaz a *regra\_pc* e a Fig. 7b não satisfaz, porque a transição B→C viola a *regra\_pc*. Figura 7a tem os códigos simbólicos positivos e negativos que respectivamente são [0,1,2] e [0,1,2].

**Regra\_pc:** Seja a ponte  $P_j$  dos estados  $A$  e  $B$  do GTR e os semi-códigos  $Sci, Sck \in B\_CLK+$  e  $Scx, Scy \in B\_CLK-$  onde  $Sc$  é um valor inteiro. Se  $P_j$  é negativo então os estados  $A$  e  $B$  devem ter o mesmo semi-código  $\in B\_CLK+$ , por exemplo:  $A=Sci\&^3Scx$  e  $B=Sci\&Scy$ . Se  $P_j$  é positivo então os estados  $A$  e  $B$  devem ter o mesmo semi-código  $\in B\_CLK-$ , por exemplo:  $A=Sci\&Scx$  e  $B=Sck\&Scx$ .

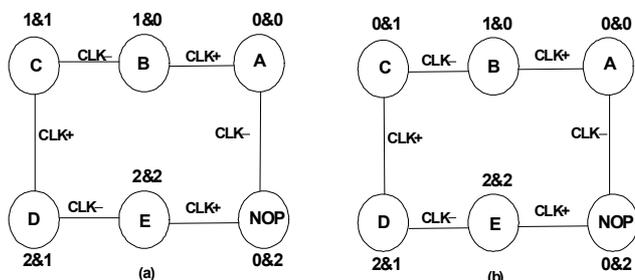


Fig.7. GTR codificado simbolicamente: a) satisfaz a *regra\_pc*; b) não satisfaz a *regra\_pc*.

**B2. Redução de código**

O algoritmo de redução de variáveis de estado procura fundir semi-códigos [21]. O algoritmo gera todas as combinações de fusão de semi-códigos (força-bruta). Figuras 8a,b,c mostram mapas de semi-códigos.

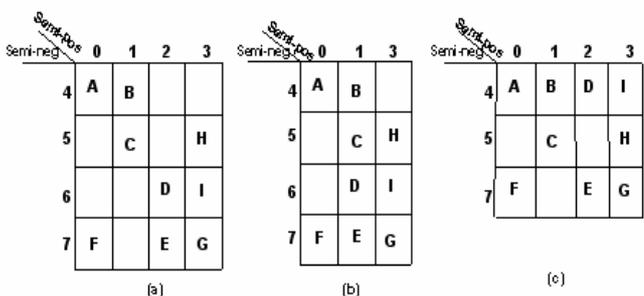


Fig.8. Mapas de semi-códigos: a) inicial; b) fusão pontes positiva; c) fusão pontes negativa.

**B3. Codificação binária**

O algoritmo proposto para codificação binária do GTR usa os conceitos de máscaras de inversão e *Snakesequence* [21]. A codificação é realizada em duas etapas. A primeira para o banco  $B\_CLK+$  e a seguir para o banco  $B\_CLK-$ .

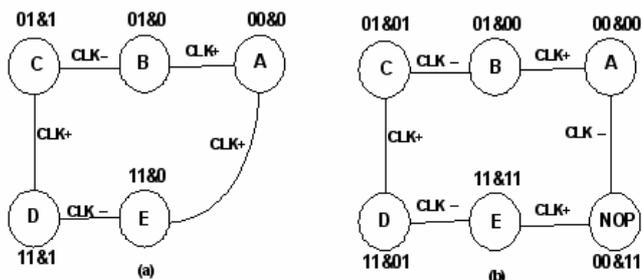


Fig. 9. GTR codificado: a) três variáveis; b) quatro variáveis.

**IV. EXEMPLO**

Nesta seção ilustramos o nosso método com o *benchmark seven.ems* [21]( ver Fig. 10). A ferramenta SYntool\_DET lê a especificação no formato Kiss2 ou no formato EMS. O segundo passo é minimização de estados. Figura 11 mostra o GTE minimizado do *seven.ems*. Os estados  $D$  e  $E$  se fundiram. O terceiro passo gera o GTR e foi usada a opção de correção de paridade do ciclo. O estado  $NOP$  foi inserido entre os estados  $H$  e  $A$  (ver Fig.12). Na etapa seguinte do terceiro passo o GTR é rotulado com o sinal de relógio ( $CLK$ ) (ver Fig. 13). O quarto passo é realizado a codificação simbólica do GTR (ver Fig. 14). A codificação simbólica resultante precisou no banco  $B\_CLK+$  de três códigos [0,1,2] e no banco  $B\_CLK-$  de quatro códigos [0,1,2,3]. A próxima etapa é a redução de código. Houve a eliminação do código 2 no banco  $B\_CLK+$ , portanto passa a ter dois códigos [0,1] (ver Fig.15,16). Na ultima etapa do quarto passo ocorre a codificação binária. Figura 17 mostra o GTR codificado. A codificação precisou de três variáveis de estado. Duas variáveis de estado para o banco  $B\_CLK-$  e uma variável para o banco  $B\_CLK+$ . O quinto e ultimo passo é a minimização lógica. Figuras 18,19,20 mostram o circuito lógico resultante.

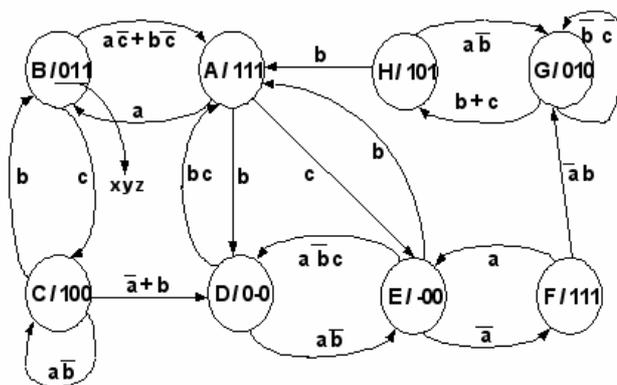


Fig. 10. GTE: benchmark *seven.ems*.

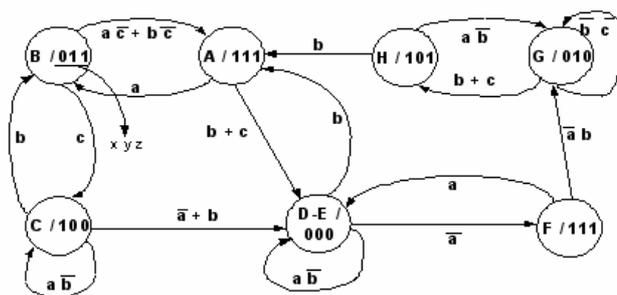


Fig. 11. GTEMIN: benchmark *seven.ems*.

<sup>3</sup> O símbolo & significa concatenação.

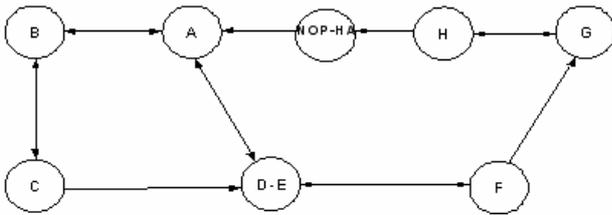


Fig. 12. GTRMIN-CP: benchmark seven.ems com correção de paridade.

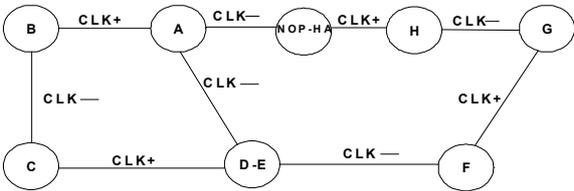


Fig. 13. GTR rotulado.

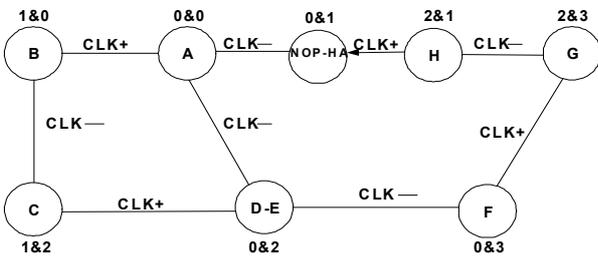


Fig. 14. GTR codificado simbolicamente.

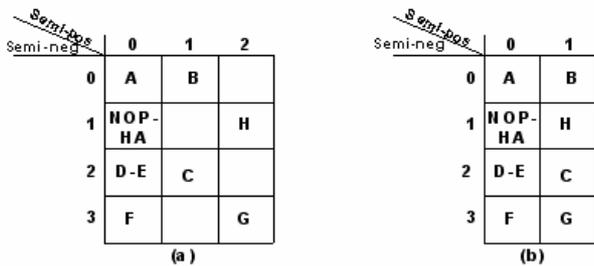


Fig. 15. Mapa de códigos simbólicos: a) inicial; b) reduzido.

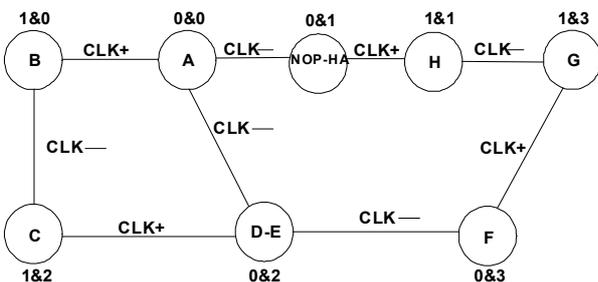


Fig. 16. GTR com código simbólico reduzido.

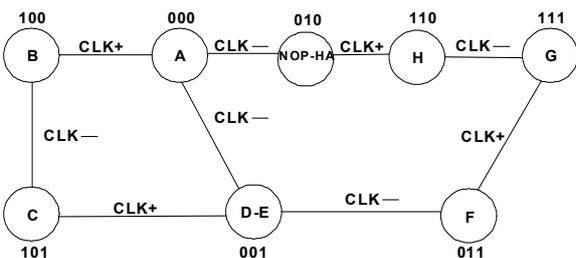


Fig. 17. GTR codificado.

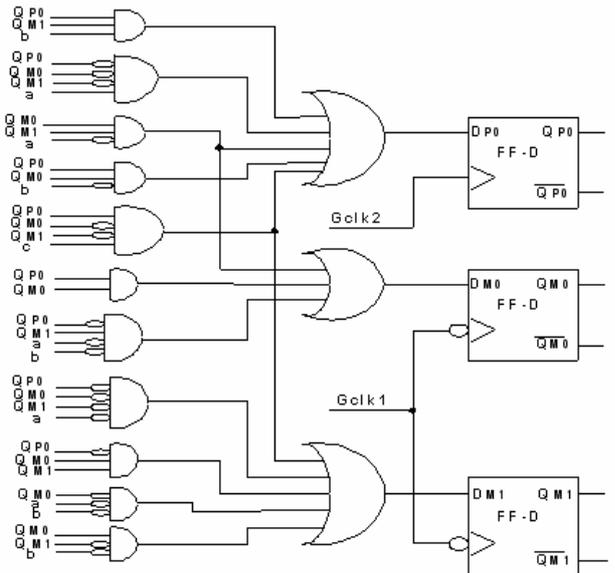


Fig. 18. Circuito lógico: lógica de excitação.

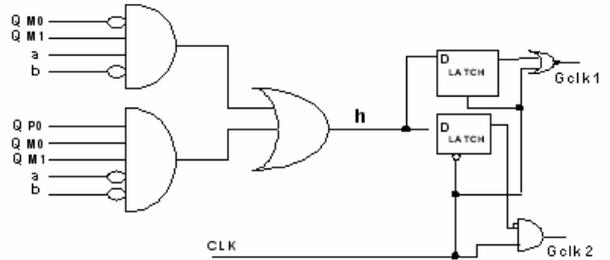


Fig. 19. Circuito lógico: lógica de inibição de relógio.

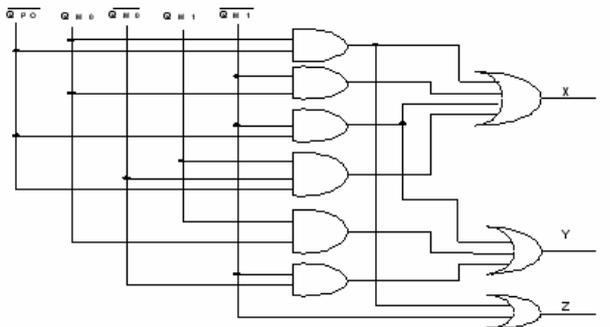


Fig. 20. Circuito lógico: lógica de saída.

## V. DISCUSSÃO E RESULTADOS

A nossa ferramenta tem duas opções de síntese automática, que são a convencional e a de redução de relógio. A síntese convencional segue o procedimento tradicional [20]. Aplicamos a nossa ferramenta em dez exemplos da literatura (benchmarks). A tabela 1 mostra os dados da especificação e dados da síntese convencional obtidos pela nossa ferramenta. Estes dados são: tempo de processamento e respectivamente relacionados com a lógica de excitação o número de literais e o número de portas. A tabela 2 mostra os dados obtidos pela síntese redução de relógio que incluem lógica de excitação e a função de inibição  $h$ . A coluna banco de flip-flops mostra a quantidade de FFs que respectivamente vão operar na borda de subida (+) e na borda de descida (-) do sinal de relógio. Comparando com a síntese convencional, o nosso método obteve uma redução de 14% no número de literais e uma penalidade de 5% no número de portas.

TABELA I  
RESULTADOS DA SÍNTESE CONVENCIONAL

Exemplos	Especificação		Síntese Convencional		
	Entradas/Saídas	Estados	Literais	Portas	Tempo Proc. (s)
Alarm	5 / 3	3	17	7	2
Complex	3 / 3	11	101	29	120
Dumbbell	2 / 2	8	23	12	2
Mark1	5 / 16	14	128	38	2
Pma	8 / 8	24	418	112	120
Seven	3 / 3	8	66	28	1
Shiftreg	1 / 1	8	51	21	2
Six	3 / 2	6	15	8	1
Three	2 / 1	3	15	8	1
Tma	7 / 6	20	267	80	120

TABELA II  
RESULTADOS DA SÍNTESE COM R\_A\_RELÓGIO

Exemplos	Especificação		Síntese com redução da atividade do relógio			
	Entradas/Saídas	Estados	Literais	Portas	Tempo Proc. (s)	Banco de Flip-Flops
Alarm	5 / 3	3	28	8	3	1+ / 1-
Complex	3 / 3	11	69	35	127	2+ / 2-
Dumbbell	2 / 2	8	23	11	4	2+ / 1-
Mark1	5 / 16	14	130	49	3	2+ / 3-
Pma	8 / 8	24	349	107	28	4+ / 4-
Seven	3 / 3	8	60	26	6	1+ / 2-
Shiftreg	1 / 1	8	43	20	1	1+ / 3-
Six	3 / 2	6	17	10	2	2+ / 1-
Three	2 / 1	3	24	12	3	1+ / 1-
Tma	7 / 6	20	218	74	185	2+ / 4-

VI. CONCLUSÃO

Neste artigo apresentamos a ferramenta Syntool\_DET que sintetiza automaticamente controladores síncronos com inibição de relógio e que operam nas duas bordas (subida/descida) da transição do sinal de relógio. Os nossos controladores somente usam FF's que são sensíveis a uma simples borda do sinal de relógio. Esta característica permite uma grande redução na atividade do relógio e conseqüentemente redução no consumo de potência. A ferramenta Syntool\_DET foi implementada em C++ tendo aproximadamente 10.000 linhas de código. Para trabalhos futuros pretendemos testar exaustivamente a ferramenta e realizar a estimação de potência para um grande conjunto de benchmarks e comparar com a síntese convencional [23,24].

REFERÊNCIAS

- [1] Li-Chuan Weng, X. J. Wang, and Bin Liu, "A Survey of Dynamic Power Optimization Techniques," Proc. Of the 3rd IEEE Int. Workshop on System-on-Chip for Real-Time Applications, pp. 48-52, 2003.
- [2] J. J. Rodriguez, at. al., "Features, Design Tools, and Applications Domains of FPGAs," IEEE Trans. On Industrial Electronics, vol. 54, no 4, pp. 1810-1823, August, 2007.
- [3] F. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," IEEE Trans. On VLSI Systems, vol. 2, no. 4, pp.446-455, December 1994.
- [4] 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.
- [5] 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.
- [6] S. Devadas and S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," Proc. 32<sup>nd</sup> ACM/IEEE DAC, pp.242-247, 1995.
- [7] Luca Benini and G. De Micheli, "Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines," IEEE Trans. on CAD of Integrated Circuits and Systems, Vol.15, No.6, pp.630-643, June 1996.
- [8] Q. Wu, M. Pedram, and X. Wu, "Clock-Gating and Its Application to Low Power Design of Sequential Circuits," IEEE Trans. On Circuits and Systems-I: Fundamental Theory and Applications, vol. 47, no.103, pp.415-420, March 2001.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] J. C. Monteiro and A. L. Oliveira, "Implicit FSM Decomposition Applied to Low-Power Design," IEEE Trans. on VLSI Systems, Vol. 10, No. 5, pp.560-565, October 2002.
- [13] B. Liu, et al., "FSM Decomposition for Power Gating Design Automation in Sequential Circuits," 76th Int. Conf. on ASIC, ASICON, pp.944-947, 2005.
- [14] P. Baccheletta et al. "Low-Power State assignment Techniques for Finite State Machines," IEEE Int. Symposium on Circuits and systems, Geneva, Switzerland, pp.641-644, May 2000.
- [15] 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.
- [16] J.-Mou Tseng and J.-Yang Jou, "A Power-Driven Two-Level Logic Optimizer," Proc. Of the ASP-DAC, pp.113-116, 1997.
- [17] H. Choi and S. Ho Hwang, "Improving Two-Level Logic Minimization Technique for Low Power Driven Multi-Level Logic Re-Synthesis," Proc. 40th Midwest Symposium on Circuits and systems, pp.1026-1029, 1997.
- [18] Internet: [www.altera.com](http://www.altera.com), 2009.
- [19] D. L. Oliveira, et al., "Synthesis of Low-Power Synchronous Controllers using FPGA Implementation," IEEE IV Southern Conference on Programmable Logic, pp.221-224, 2008.
- [20] R. H. Katz, Contemporary Logic Design, The Benjamin/ Cummings Publishing Company, Inc., 2<sup>a</sup> edition 2003.
- [21] Jozias Del Rios, et. al., "Automação do Projeto de Circuitos Controladores Síncronos de Baixa Potência," Relatório Técnico - ITA Junior, 2008.
- [22] E. J. MacCluskey, Logic Design Principles With Emphasis on Testable Semicustom Circuits, Prentice-Hall, 1986
- [23] J. H. Anderson and F. N. Najm, "Power Estimation Techniques for FPGAs," IEEE Trans. On VLSI Systems, vol. 12, no. 10, pp.1015-1027, October, 2004.
- [24] J. C. Monteiro, Techniques for Power Management at the Logic Level. IEEE Int. Conf. on Electronic Circuits and Systems, pp.181-184, 1998.