

# Síntese de Controladores Síncronos de Baixo Consumo

Duarte Lopes de Oliveira

Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – IEEA – ITA  
Praça Marechal Eduardo Gomes, 50 – CEP 12228-900 – São José dos Campos– São Paulo – Brazil.

**Resumo** — Na atualidade, muitos sistemas digitais são descritos por uma arquitetura composta de uma rede de controladores síncronos e *datapaths*. Eles são alimentados por bateria e são implementados na tecnologia VLSI e/ou FPGAs (*Field Programmable Gate Array*). Como a bateria deve ter uma vida longa útil, a redução do consumo de energia é a tarefa mais importante no projeto destes sistemas. Para reduzir a potência dissipada, diversas estratégias foram propostas na literatura, seja para o controlador, como para o *datapath*. Neste artigo propomos um método que aplica uma nova estratégia para síntese de controladores síncronos de baixo consumo. O nosso método sintetiza controladores síncronos que operam nas duas bordas da transição do sinal de relógio, mas usa somente flip-flops que operam em uma única borda do relógio.

**Palavras-chaves** — Controladores síncronos, síntese de baixo consumo, controle lógico do relógio, flip-flops operando nas transições do relógio.

## I. INTRODUÇÃO

Com o avanço da microeletrônica, cada vez mais sistemas digitais de alta complexidade são projetados. Muitos destes sistemas têm como característica comum serem alimentados por bateria e estão voltados para diferentes aplicações, tais como: comunicação sem fio, computador portátil, aeroespacial (satélite, mísseis, etc), aviação, automobilística, médica, etc. Por serem alimentados por bateria é desejável que ela tenha uma vida longa útil, portanto, a dissipação de potência é um parâmetro muito importante no projeto destes sistemas [1,2]. Estes sistemas podem ser implementados na tecnologia VLSI e/ou em FPGAs (*Field Programmable Gate Array*). FPGAs têm se tornado o meio popular para implementar circuitos digitais. A tecnologia FPGA cresceu enormemente nos últimos anos, gerando FPGAs até com 50 milhões de portas, permitindo com que sistemas digitais complexos possam ser programados nestes dispositivos [3].

Tradicionalmente, os circuitos digitais são implementados com componentes que são construídos com a tecnologia CMOS. A potência dissipada nos componentes CMOS segue a seguinte expressão [4]:

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

Onde:  $P_{TM}$  é a potência total média dissipada,  $V_{DD}$  é a tensão de alimentação,  $f$  é a frequência de operação, o fator  $N$  é atividade de chaveamento, isto é, o número de transições na saída de uma porta, e o fator  $Q_{sc}$  e  $C$  são respectivamente a quantidade de carga e a capacitância [4].

Na equação (1), o primeiro termo representa a potência dissipada dinâmica. O segundo termo representa a potência dissipada relacionado com a corrente de curto. O terceiro termo representa a potência dissipada estática relacionada com a corrente de fuga. Na tecnologia CMOS estática a maior fração da potência dissipada ocorre durante a atividade de chaveamento dos eventos (potência dinâmica) [4]. A dissipação da potência média na porta  $g$  pode ser simplificada ao primeiro termo de (1):

$$P_{\text{média-g}} = 1/2 \cdot C_g \cdot V_{DD}^2 \cdot f \cdot N_{\text{média-g}} \quad (2)$$

A síntese de máquinas de estado finito (MEF) tem um papel importante no projeto de circuitos digitais alimentados por bateria. Muitos circuitos digitais são descritos por uma arquitetura composta de uma rede de controladores + *datapaths* e/ou processadores [5]. Os controladores síncronos de tais circuitos são muitas vezes especificados na forma de uma MEF, que consistem de vários estados e transições entre estados, isto é são especificados por um grafo de transição de estados (GTE).

As técnicas de redução da potência dinâmica são aplicadas nos diferentes níveis do projeto digital [1,2]. Na síntese de controladores síncronos as propostas de redução de potência estão sendo feitas no nível lógico que são: controle lógico do relógio (*gated-clock*) [6,7], *Flip-Flops* que atuam nas duas bordas da transição do relógio [8,9,10], decomposição [11,12], assinalamento de estados [13,14,15] e minimização lógica [16,17,18].

Em um sistema digital, a parte seqüencial é a maior contribuidora na dissipação de potência. Recentes estudos indicam que o relógio destes sistemas consome uma grande porcentagem (15% a 45%) da potência do sistema [1,2]. Assim, a potência do circuito, pode ser grandemente reduzida se reduzirem à atividade do relógio.

Nas propostas citadas, para reduzir a potência dissipada dinâmica dos controladores síncronos, as duas primeiras são muito interessantes:

### A. Controle lógico do relógio

Esta estratégia usa uma lógica adicional, para desabilitar o relógio nos estados com auto-transição (*self-loop*). Para algumas MEF, a maioria dos ciclos do relógio são usados nos estados com auto-transição. Nestes estados também há dissipação de potência dinâmica, porque internamente os FFs mudam de estado, embora não haja nenhuma mudança de valor na saída dos FFs. A fig. 1 mostra a arquitetura alvo para controladores com inibição de relógio [6].

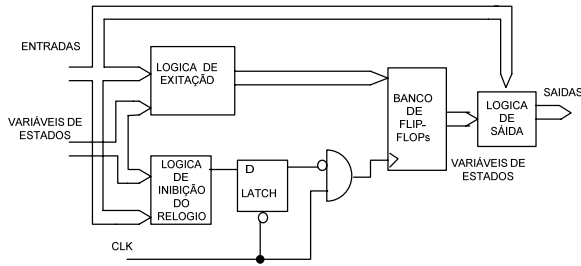


Fig. 1. Controle lógico do relógio.

### B. Flip-Flop gatilhado nas duas bordas do relógio

Os controladores usam *flip-flops* que operam nas bordas de subida e descida da transição do sinal de relógio (*double-edge triggered* - DET) [9]. Para uma taxa de processamento de dados (*data throughput*) os DET-FFs necessitam da metade da taxa de relógio, portanto reduz a atividade do relógio. Comparando os DET-FFs com FFs que atuam em uma única borda da transição do sinal de relógio (*single-edge triggered* - SET) há aumento do consumo de potência e/ou área (número de transistores) [9]. Dois são os problemas nesta estratégia: 1) a maioria das bibliotecas VLSI *Standard cell* não contém este tipo de FF e as macro-células das FPGAs usam SET-FF D [3]; 2) não há uma arquitetura que desabilita o relógio nas auto-transições de uma MEF que usa DET-FF. Para ilustração, a fig. 2 mostra um esquema lógico clássico de um DET-FF D.

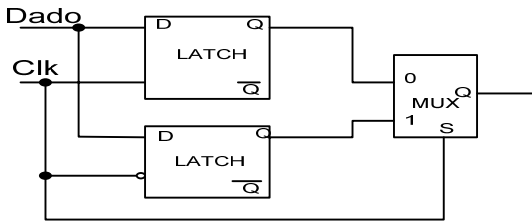


Fig. 2. Esquema lógico DET-FF D.

Neste artigo propomos um simples procedimento para síntese de controladores síncronos que reduz drasticamente a atividade do sinal de relógio. A redução da atividade do sinal de relógio é obtida através de duas estratégias. Na primeira estratégia, o nosso método sintetiza controladores síncronos, que operam nas duas bordas do sinal de relógio, mas usando SET-FFs. Na segunda estratégia, propomos uma arquitetura alvo que desabilita o sinal de relógio que opera nas duas bordas, os estados com auto-transição. A fig. 3 mostra a arquitetura alvo modelo Mealy usada para implementar os nossos controladores síncronos. No modelo Moore, o bloco da lógica de saída não contém os sinais de entrada.

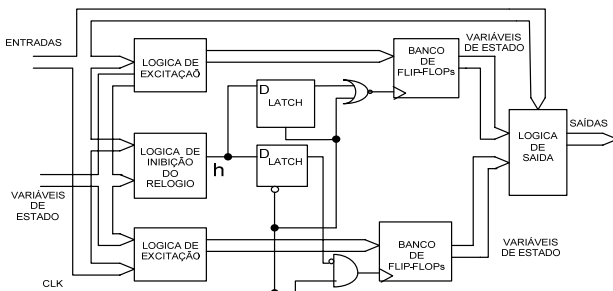


Fig. 3. Arquitetura alvo proposta.

O restante deste artigo está organizado como segue. Na seção II apresentamos o nosso método; na seção III ilustramos o nosso método com um exemplo real; na seção IV discutimos as vantagens e limitações do nosso método.

## II. PROCEDIMENTO

O nosso método segue os passos tradicionais da síntese de controladores síncronos. A metodologia é composta de 6 passos, e parte da especificação GTE:

- 1: Realizar a minimização de estados do GTE gerando o  $GTE_{MIN}$ . Diversos algoritmos podem ser usados, por exemplo, o algoritmo em [19]
- 2: A partir do  $GTE_{MIN}$  gerar o grafo de transição do relógio (GTR).
- 3: A partir do GTR obter o número máximo de variáveis de estado, onde:  $N_{VAR-CLK-} = \lfloor \log_2 N_{TE-CLK-} \rfloor$  e  $N_{VAR-CLK+} = \lfloor \log_2 N_{TE-CLK+} \rfloor$ . Sendo  $N_{TE-CLK+}$  e  $N_{TE-CLK-}$  respectivamente o número de transições de estado com as transições do sinal de relógio  $clk+$  e  $clk-$ , e  $N_{VAR-CLK+}$  e  $N_{VAR-CLK-}$  respectivamente o número de variáveis de estado que cobrem as transições de estado do GTR. O número máximo de variáveis de estado ( $N_{Tvar}$ ) é  $N_{Tvar} = N_{VAR-CLK+} + N_{VAR-CLK-}$ .
- 4: Usando  $N_{VAR-CLK+}$  e  $N_{VAR-CLK-}$  realizar o assinalamento de estados do conjunto de transições de estado do GTR.
- 5: Otimizar o código de estados do GTR, gerando o  $GTE_{MIN}$  codificado.
- 6: Realizar a minimização lógica para o  $GTE_{MIN}$  codificado obtendo as equações de excitação, as equações de inibição e as equações de saída. Os estados que pertencem as auto-transições são *don't-care*. Diversos algoritmos podem ser usados, por exemplo, o algoritmo de Quine-McCluskey [19,20].

O nosso método gera no segundo passo um grafo que descreve em que borda (descida ou subida) o sinal de relógio realizará a transição de estado. Neste passo, um algoritmo de força bruta, otimiza o número de bordas, para percorrer todas as transições de estado do GTE, a partir do seu estado inicial. Como ilustração, a fig. 4a mostra um GTE onde as entradas e saídas estão omitidas. A fig. 4b mostra um possível assinalamento de transições do relógio (GTR) para o GTE da fig. 4a. Um estado do tipo NOP (sem operação) foi introduzido entre a transição de estado  $F \rightarrow B$  do GTE para gerar um ciclo completo.

Nos passos 3,4 e 5, o método usa o GTR obtido para codificar os estados do GTE. A codificação dos estados é realizada em duas etapas. Na primeira etapa, usando o número máximo de variáveis de estado, um algoritmo codifica os estados seguindo as transições de estado e o menor número de chaveamentos. Na segunda etapa um algoritmo de força bruta procura otimizar o número de variáveis.

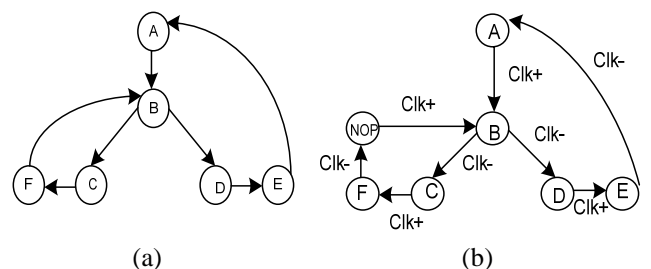


Fig. 4. Especificações: a) GTE; b) GTR.

### III. EXEMPLO

Para mostrarmos o nosso método, escolhemos uma simples função de um automóvel [21]. Este exemplo é um sistema de controle de alarme para cinto de segurança. Ele é composto por dois controladores que operam concorrentemente. Um dos controladores é um temporizador. A fig. 5 mostra este sistema de alarme proposto em [21]. Numa linguagem natural, este sistema é especificado por um projetista, como: depois de 5 segundos, que a chave de ignição do automóvel é ligada, se o cinto não está posto corretamente, um alarme é acionado por cinco segundos ou até que a chave é desligada. As fig. 6 e 8 mostram respectivamente os GTEs minimizados do controlador e do temporizador. O segundo passo gera os GTRs do controlador e do temporizador. As fig. 7 e 9 mostram respectivamente  $GTR_{controlador}$  e  $GTR_{temporizador}$ . Um estado NOP foi introduzido nas duas MEFs. O terceiro e quarto passo gera os GTEs não otimizados. As fig. 10a,b mostram respectivamente os códigos de estados não otimizados do temporizador (seis variáveis de estado) e do controlador (duas variáveis de estado). O quinto passo há uma otimização de variáveis de estado; o temporizador necessita de quatro variáveis de estado e o controlador de duas variáveis de estado. As fig. 11a,b mostram a otimização dos códigos. As fig. 12 e 13 mostram respectivamente os  $GTE_{controlador}$  e  $GTE_{temporizador}$  codificados. O sexto passo é realizada a minimização lógica. As fig. 14,15,16 mostram respectivamente os circuitos lógicos do controlador, o seu inibidor do relógio e o temporizador com o seu inibidor.

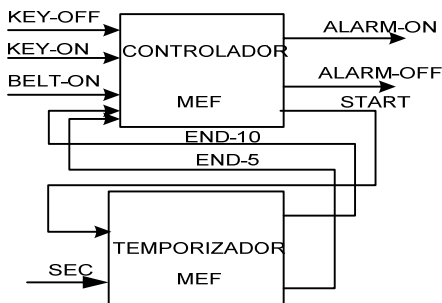


Fig. 5. Exemplo: sistema de alarme de cinto de segurança [21].

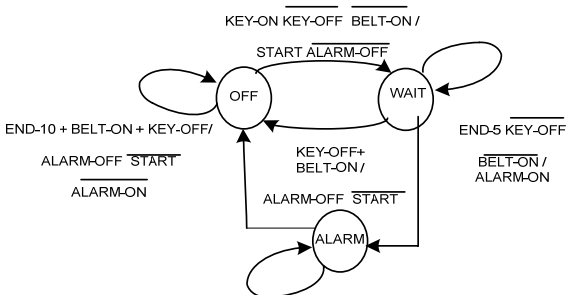


Fig. 6. Especificação:  $GTE_{controlador}$

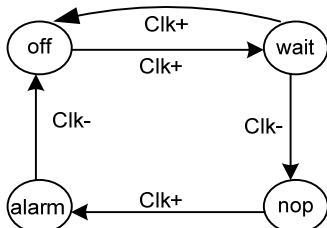


Fig. 7. Especificação:  $GTR_{controlador}$

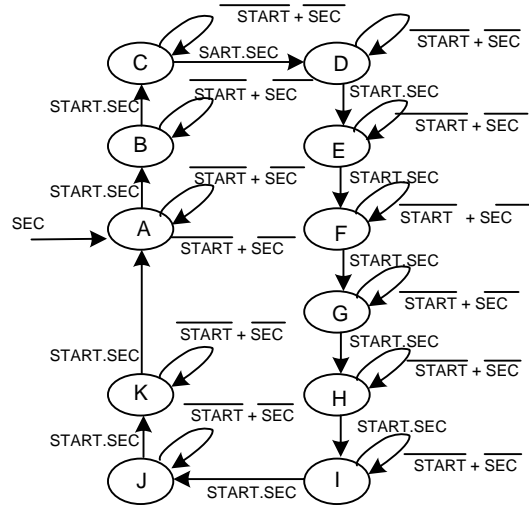


Fig. 8. Especificação:  $GTE_{temporizador}$

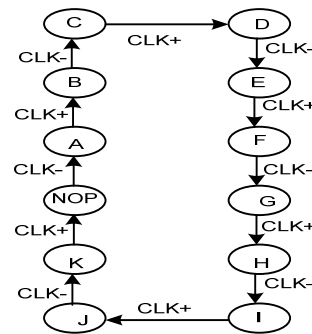


Fig. 9. Especificação:  $GTR_{temporizador}$

ESTADOS	CLK+		CLK-	
	Q0	Q1	Q2	Q3
A	000	000		
B	100	000		
C	100	100		
D	110	100		
E	110	110		
F	010	110		
G	010	010		
H	011	010		
I	011	011		
J	111	011		
K	111	111		
NOP	000	111		

ESTADOS	CLK+		CLK-	
	Q0	Q1	Q0	Q1
OFF	0	0		
WAIT	1	0		
NOP	1	1		
ALARM	0	1		

(a) (b)

Fig. 10. Codificação de estados não otimizado: a) temporizador; b) controlador

ESTADOS	CLK+		CLK-	
	Q0	Q1	Q2	Q3
A	00	00		
B	10	00		
C	10	10		
D	00	10		
E	00	11		
F	10	11		
G	10	01		
H	11	01		
I	11	00		
J	01	00		
K	01	01		
NOP	00	01		

ESTADOS	CLK+		CLK-	
	Q0	Q1	Q0	Q1
OFF	0	0		
WAIT	1	0		
NOP	1	1		
ALARM	0	1		

(a) (b)

Fig. 11. Codificação de estados otimizados: a) temporizador; b) controlador

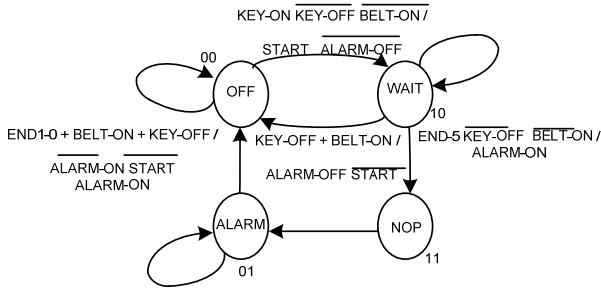


Fig. 12. GTE codificado: controlador.

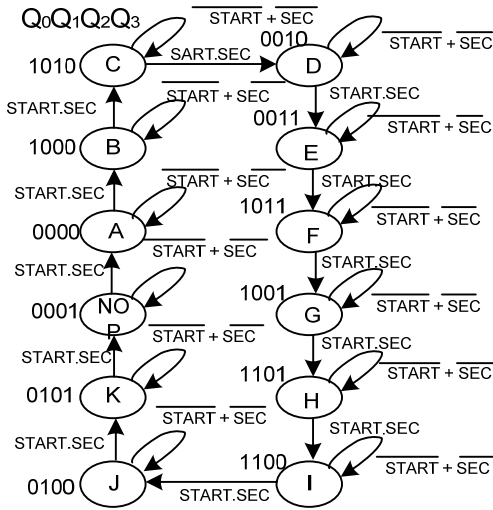


Fig. 13. GTE codificado: temporizador.

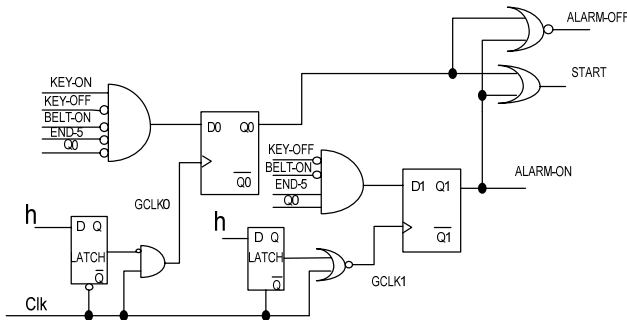


Fig. 14. Circuito lógico: controlador.

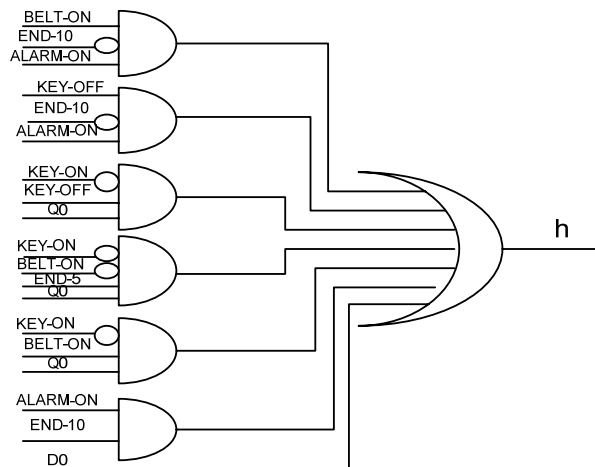


Fig. 15. Circuito lógico: inibidor do relógio

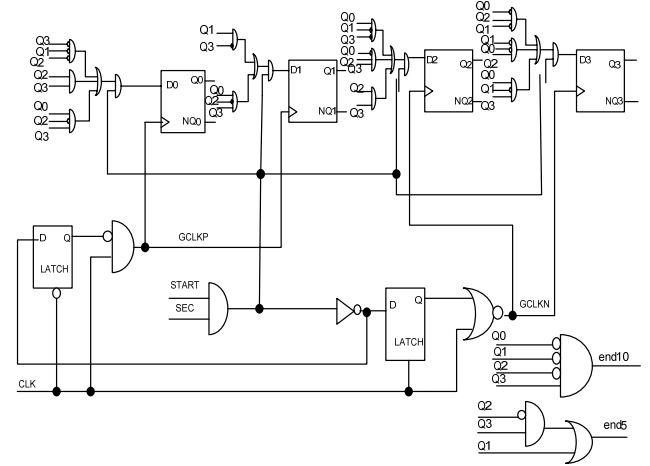


Fig. 16. Circuito lógico: temporizador

#### IV. DISCUSSÃO E RESULTADOS

Inicialmente, discutiremos as vantagens da arquitetura alvo e do método proposto. Posteriormente discutiremos algumas limitações. O exemplo da seção III foi simulado no Altera-Max-Plux-II versão 10.2. A fig. 17 mostra a simulação do controlador percorrendo todas as transições de estado, com um número reduzido da atividade do relógio. A fig. 18 mostra a simulação do temporizador que percorre todo o ciclo de transições com uma taxa reduzida de relógio. Analisando o requisito de área (número de transistores), o nosso circuito necessitou de 354 transistores. Este circuito sintetizado pelo método clássico [19,20], necessitou de 341 transistores, portanto houve apenas um aumento de área de 4%.

Comparando com o método clássico, citamos três vantagens do nosso método:

- 1) O método proposto decompõe o GTE em dois sub-grafos, portanto, pode facilitar a tarefa de assinalamento de estados que é um problema NP-difícil. A complexidade desta tarefa está relacionada com o número de estados [19].
- 2) Como o sinal de relógio opera nas duas bordas, podemos reduzir a taxa do relógio pela metade que continuaremos com a mesma velocidade de processamento, portanto há uma redução na atividade do sinal de relógio.
- 3) A arquitetura proposta desabilita o sinal relógio nos estados com auto-transição, portanto há uma redução na atividade do sinal de relógio.

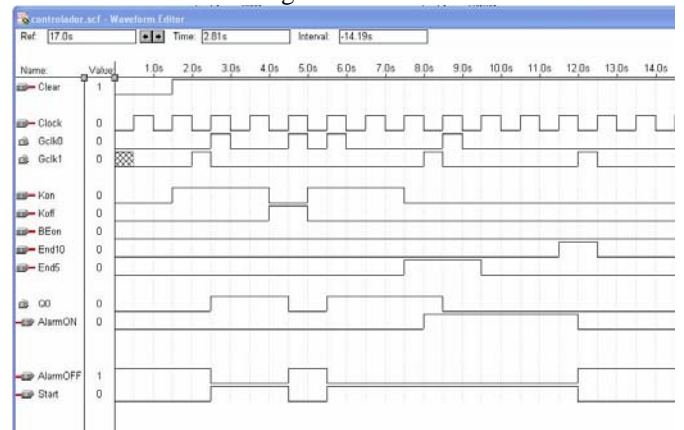


Fig. 17. Simulação: controlador da fig. 14 e 15.

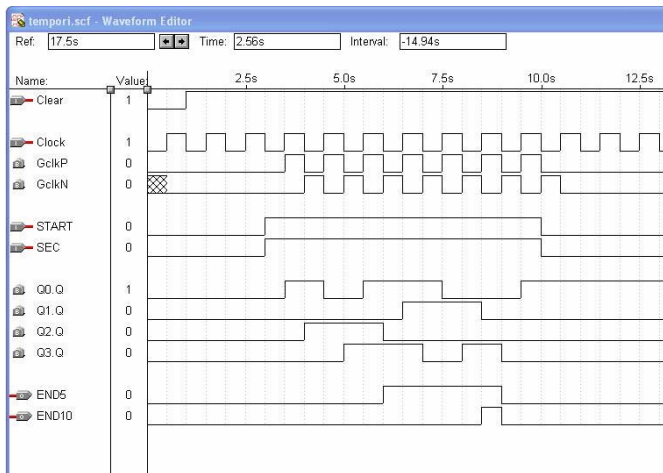


Fig. 18. Simulação: temporizador da fig. 16.

O método proposto possui duas limitações: 1) A síntese registradores de deslocamentos e contadores podem gerar circuitos ineficientes; 2) Para aplicações, com muitos ciclos e que contém um número ímpar de transições de estado, o nosso método é ineficiente. A fig. 19 mostra um GTE que para gerarmos o seu GTR devemos inserir muitos estados NOP. Este aumento de estados acarreta uma degradação no desempenho (área, velocidade e consumo de energia) no circuito final sintetizado.

O nosso método é colocado como uma alternativa de projeto. Ele pode ser usado em aplicações em que efetivamente há uma redução da potência dissipada e a degradação de área e velocidade de algumas transições de estado não comprometam a viabilidade do projeto.

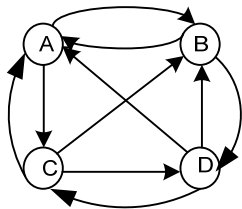


Fig. 19. Especificação GTE.

## V. OBSERVAÇÕES FINAIS

Neste artigo apresentamos a síntese de controladores síncronos, cujo parâmetro mais importante é o consumo de potência. Na síntese de controladores de baixo consumo, diversas estratégias foram propostas nas duas últimas décadas. Neste artigo propomos uma nova estratégia, e acreditamos que para muitas aplicações podemos ter bons resultados. Como trabalho futuro, propomos uma ferramenta para síntese automática e estimação de potência dissipada voltada para FPGA [22].

## REFERÊNCIAS

[1] S. Devadas and S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits", Proc. 32nd ACM/IEEE DAC, pp.242-247, 1995.  
 [2] 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.  
 [3] Rodriguez, J. J., et al., "Features, Design Tools, and Applications Domains of FPGAs", IEEE Trans. On Industrial Electronics, vol. 54, no 4, pp. 1810-1823, August, 2007.  
 [4] 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.  
 [5] Jozwiak, L. 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.  
 [6] 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.  
 [7] 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.  
 [8] 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.  
 [9] 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.  
 [10] 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.  
 [11] 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.  
 [12] B. Liu, et al., "FSM Decomposition for Power Gating Design Automation in Sequential Circuits", 76th Int. Conf. on ASIC, ASICON, pp.944-947, 2005.  
 [13] M. Koegst, et al. "Low Power Design of FSMs by State Assignment and Disabling Self-Loops", Proc. 23rd EUROMICRO "new Frontiers of Information Technology", pp. 323-330, 1997.  
 [14] M. Koegst et al., Multi-Criterial State Assignment for Low Power FSM Design. Proc. 24th EUROMICRO Conference, pp.261-268, 1998.  
 [15] 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.  
 [16] S. Iman and M. Pedram, "Two-level Logic Minimization for Low Power", IEEE/ACM Conf. Int. on CAD Digest of Technical Papers, pp.433-438, 1995.  
 [17] R. I. Bahar and F. Somenzi, Boolean Techniques for Low Power Driven Re-Synthesis, IEEE/ACM Conf. Int. on CAD Digest of technical Papers, pp.428-432 1995.  
 [18] J.-Mou Tseng and J.-Yang Jou, A Power-Driven Two-Level Logic Optimizer. Proc. Of the ASP-DAC, pp.113-116, 1997.  
 [19] E. J. MacCluskey, *Logic Design Principles With Emphasis on Testable Semicustom Circuits*, Prentice-Hall, 1986  
 [20] R. H. Katz, *Contemporary Logic Design*, The Benjamin/Cummings Publishing Company, Inc., 2a edition 2003.  
 [21] 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.  
 [22] 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.