

CONVERSOR ANALÓGICO-PARA-DIGITAL IMPLEMENTADO POR UMA ARQUITETURA COMPUTACIONAL ASSÍNCRONA¹

Leo Weber

e-mail: leoweber@uol.com.br

Engenheiro Eletricista, Especialista em Controle de Processos Industriais, Doutorando em Informática pela Universidade das Ilhas Baleares (UIB) – Espanha. Artigo revisado pelo Dr. Walter Del Picchia (Escola Politécnica da USP).

Resumo: O propósito deste artigo é demonstrar o emprego de uma arquitetura computacional assíncrona na implementação de um conversor analógico-para-digital por aproximação sucessiva.

Palavras chave – Arquitetura de computadores. Lógica assíncrona. Sistemas digitais sem relógio. Conversor analógico-para-digital.

Abstract: The purpose of this article is to demonstrate the use of an asynchronous computer's architecture in the implementation of an successive approximation analog-to-digital converter.

Key words – Computer's architecture. Asynchronous logic. Digital systems without clock. Analog-to-digital converter.

1. Introdução

A idéia de projetar computadores digitais com lógica assíncrona esteve presente desde os primeiros sistemas. O Modelo Geral de um Sistema Digital, discutido por Phister, relaciona saídas e entradas por um sistema simultâneo de funções digitais [PHIS]. Sua concepção apresenta um modelo que pode ser representado adequadamente por uma caixa preta, incluindo sistemas digitais ou circuitos de chaveamento seqüenciais.

Há várias concepções de projeto com lógica assíncrona, nas quais inúmeros grupos de pesquisa atuam simultaneamente, como, só para citar, [DAVI], [EBER], [NOWI] e [SUTH]. Uma arquitetura assíncrona passível de implementação é a baseada em uma memória [MART] [MYER] [TIND] [WEB3]. Sua simplicidade, aliada a uma redução drástica da velocidade, gera menor ocupação de espaço com baixo custo. Sendo um sistema em tempo real, isto é, sem um sincronismo próprio, tem grande flexibilidade de programação, pois, basta alterar as suas características para aproveitar o circuito para outra função totalmente diferente. A nomenclatura deste tipo de máquina é “n-m-p” [CUES] [MART], onde:

n = variáveis booleanas independentes (entradas)

m = variáveis booleanas dependentes (saídas)

p = variáveis booleanas internas (realimentação ou memórias)

A diferença entre a arquitetura von Neumann e a assíncrona tipo “n-m-p”, não-von Neumann, é que a primeira é baseada em microprocessador e a segunda é um computador sem microprocessador, sem controle, sem centralização e onde o que é informado na entrada decide o destino da próxima saída, pois apresenta uma percepção determinística na entrada [PESO] [WEB2].

A Transformada Numérica, também denominada TN, é uma ferramenta matemática usada para descrever funções e expressões da álgebra booleana e do cálculo proposicional, onde cada expressão é definida por seqüências numéricas determinadas através de regras estabelecidas, operando, portanto, no campo numérico. Entre as diferentes aplicações nas quais a transformada

¹ Pesquisa financiada pela Fapergs (Fundação de Amparo à Pesquisa do Estado do RS).

numérica pode ser empregada, existe a aplicação em simplificação de circuitos lógicos, o que possibilita a resolução de equações booleanas que descrevem o problema proposto e facilita sua implementação. Todo este processo está descrito detalhadamente em [DELP], [WEB1] e [WEB4].

2. Conversor A/D por Aproximação Sucessiva

O conversor analógico-para-digital (A/D) por aproximação sucessiva, como o próprio nome sugere, opera através de comparações que iniciam por um valor pré-determinado e, através de comparações sucessivas, encontra o dado equivalente ao valor analógico de entrada [TOCI].

Para analisar o funcionamento do conversor A/D por aproximação sucessiva, tomamos o próprio projeto, que consiste em um conversor com intervalo de 0V a 15V e um degrau de tensão de 1V. Deste modo, obtém-se 16 níveis e são necessários 4 bits, que geram a relação da Tabela 1.

Tensão (V)	bit ₃	bit ₂	bit ₁	bit ₀	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	3°
3	0	0	1	1	4°
4	0	1	0	0	2°
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	1°
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

Tabela 1: Relação entre a tensão analógica e a saída digital.
O bit₃ é o mais significativo.

O nível de tensão que se encontra no centro é o de 8V, sendo o ponto de partida para a comparação com o sinal analógico de entrada. Os próximos níveis de comparação, que são definidos pelo valor de entrada (V_{in}), têm os bits remanejados de forma a resultar em um dado de saída correspondente ao valor analógico de entrada. Se a entrada V_{in} for menor que a tensão com a qual está sendo comparada ($V_{in} < V_{comp}$), então torna-se o bit respectivo igual a 0 e o próximo igual a 1. Se $V_{in} > V_{comp}$, mantém-se o bit em questão e coloca-se o próximo bit em 1.

Por exemplo, se tivermos um valor de entrada igual a 3,7V, o conversor compara com 8V (1000 - 1°), verifica que é menor, torna o bit₃=0 e o bit₂=1, ficando desta maneira com 4V (0100 - 2°). Uma nova comparação é realizada e, como 4V ainda é maior que 3,7V, deve-se tornar o bit₂=0 e o bit₁=1. Em seguida tem-se 2V (0010 - 3°), que é menor que 3,7V, e o sistema mantém bit₁=1 e coloca bit₀=1. Desta forma, tem-se 3V (0011 - 4°). Como o conversor possui um degrau de 1V, o valor de saída correspondente aos 3,7V de entrada, será 3V. Observe que, neste caso, o sistema ainda considera $V_{in} = 3,7V$ maior que 3V. Como anteriormente já foi verificado que este valor é menor do que 4V e maior do que 2V, logicamente, para o sistema, a tensão de saída só poderá ser 3V.

3. A Arquitetura Computacional Assíncrona

A arquitetura computacional assíncrona, de forma simplificada, pode ser implementada pelo bloco II da Figura 1, através da microprogramação de memórias para a construção dos circuitos digitais realimentados. As condições pré-estabelecidas para a sua estruturação são [MART] [WEB2]:

→ O número de variáveis de realimentação “ p ” define o número de Estados Internos Estáveis do sistema, através da condição $EIE = 2^p$. Sendo assim, ou se possui o número de realimentações ou o de EIE;

→ A inserção do passo temporal “ X_0 ”, que possibilita o controle síncrono, a uma frequência elevada e regular, ou um controle assíncrono com frequência irregular e aleatória, típica condição de operações manuais.

A metodologia empregada para desenvolver o projeto e implementação da arquitetura assíncrona proposta apresenta os passos ou etapas a seguir:

- 1º. Passo) Tabela dos EIE
- 2º. Passo) Diagrama de Blocos
- 3º. Passo) Matriz de Projeto
- 4º. Passo) Tabela Verdade Equivalente
- 5º. Passo) Mapa de Gravação da Memória (Eprom, por exemplo)
- 6º. Passo) Esquema eletrônico

O diagrama de blocos poderá ser construído antes da tabela dos EIE, já que essa ordem depende das informações conhecidas inicialmente.

4. Conversor A/D Implementado por uma Arquitetura Computacional Assíncrona

Na Figura 1 é possível identificar o diagrama de blocos do conversor A/D através de um sistema assíncrono. Sua estrutura básica é composta por: I - bloco comparador, II - memória Eprom e III - conversor digital-para-analógico (D/A).

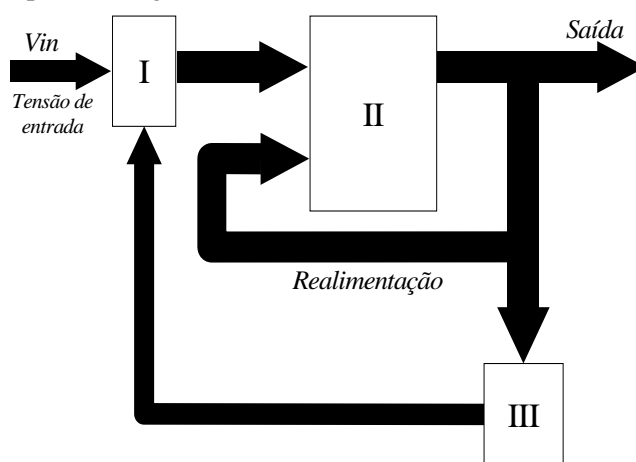


Figura 1: Diagrama de Blocos do Conversor A/D Assíncrono.

Seu funcionamento consiste em comparar a tensão de entrada V_{in} (bloco I) com a tensão enviada pelo conversor D/A (bloco III), que por sua vez recebe os dados da memória (bloco II). A realimentação dos dados de saída para a entrada do bloco II compõe o novo endereçamento, que possibilitará o acesso aos dados gravados na memória e irá gerar uma nova comparação e/ou finalização das comparações.

No caso em discussão tem-se, como informações de partida para o projeto, a faixa de variação da tensão do conversor A/D, que é de 0V a 15V, e o seu degrau de tensão, que é de 1V. Todas as explicações oferecidas anteriormente sobre o conversor A/D por aproximação sucessiva são válidas aqui e há 16 níveis como indicado na Tabela 1. Ainda, adicionalmente, tem-se uma

variável de entrada X, que é saída do comparador (bloco I), e o passo temporal X_0 . Assim, a princípio, os dados são:

- EIE = 16 (níveis), logo, $p=4$;
- Variáveis de realimentação = 4;
- Variáveis de entrada X e X_0 ;
- Variáveis de Saída = 4;

4.1. Tabela dos EIE

EIE	X=0	X=1
0	0	0
1	0	1
2	1	3
3	2	3
4	2	6
5	4	5
6	5	7
7	6	7
8	4	12
9	8	9
10	9	11
11	10	11
12	10	14
13	12	13
14	13	15
15	14	15

Tabela 2: Tabela dos EIE

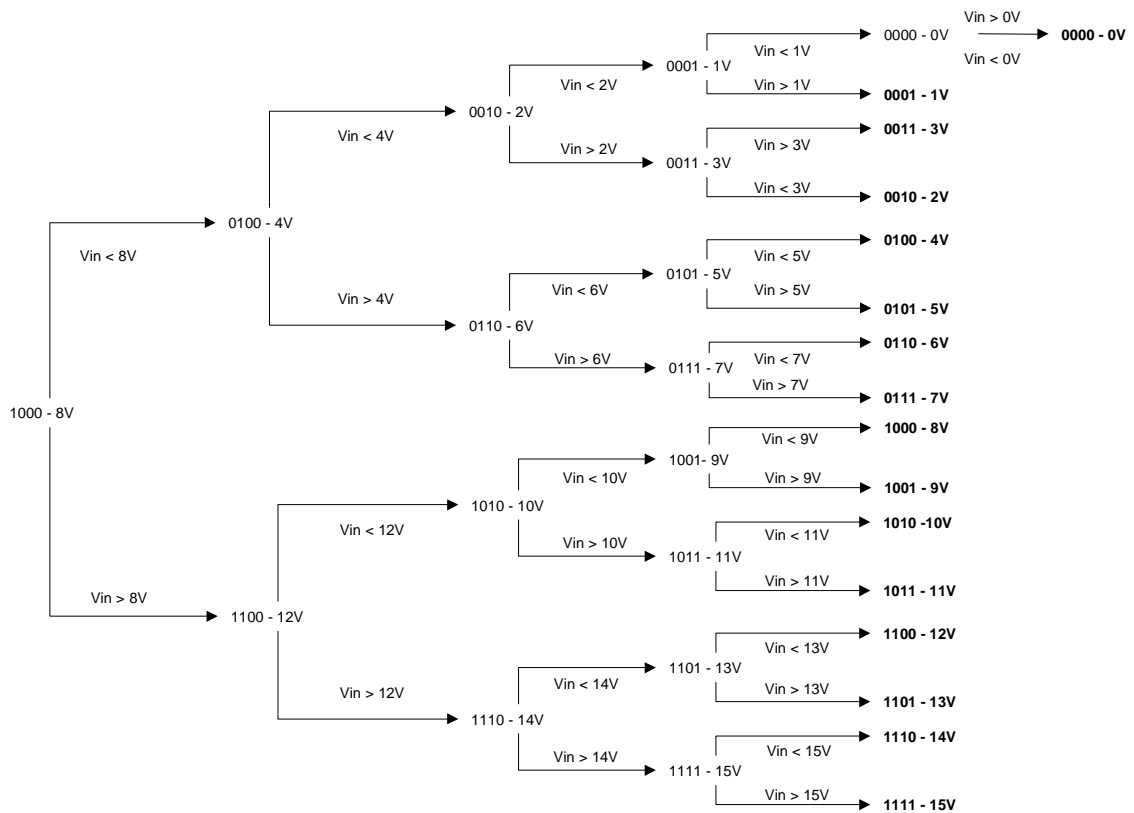


Figura 2: Estrutura hierárquica do conversor A/D

A Tabela 2 mostra os EIE e a Figura 2 apresenta graficamente a estrutura hierárquica equivalente à tabela dos EIE e demonstra mais claramente os caminhos que o conversor A/D segue durante suas comparações.

4.2. Matriz de Projeto

O projeto foi concebido considerando 16 estados internos estáveis. Porém, durante sua execução, observou-se que com os 16 EIE não era possível estruturar o sistema, porque essa condição causava "conflitos". Ou seja, diferentes situações de comparação ocupariam a mesma época matemática (t), ou ainda, existiam dois dados para um mesmo endereço, o que fisicamente não é possível.

A solução encontrada foi aumentar o número de variáveis de realimentação para 5, o que possibilitou gerar mais 16 EIE, além dos anteriores. Listando as novas condições tem-se:

→ Variáveis de realimentação = 5 ($Y_4 Y_3 Y_2 Y_1 Y_0$);

→ EIE = $2^5 = 32$ (níveis);

→ Variáveis de entrada = 2 ($X X_0$);

→ Variáveis de Saída = 4 ($Q_3 Q_2 Q_1 Q_0$) + 2 ($Q_5 Q_4$);

Os dados que preenchem a matriz de projeto (Tabela 3) são os resultados das comparações que foram executadas para cada valor de tensão entre 0 e 15V e para as condições que extrapolam os limites, como é o caso de valores de entrada menores que 0V e maiores que 15V.

EIE	$X X_0 (00)$	$X X_0 (01)$	$X X_0 (10)$	$X X_0 (11)$	
0	0 0000	24i 01 1000 t ₀₀	00e 00 0000 t ₀₁	00e 00 0000 t ₀₂	00i 10 0000 t ₀₃
1	0 0001	00i 10 0000 t ₀₄	00i 10 0000 t ₀₅	00i 10 0000 t ₀₆	01e 00 0001 t ₀₇
2	0 0010	00i 10 0000 t ₀₈	00i 10 0000 t ₀₉	02e 00 0010 t ₁₀	00i 10 0000 t ₁₁
3	0 0011	00i 10 0000 t ₁₂	00i 10 0000 t ₁₃	00i 10 0000 t ₁₄	03e 00 0011 t ₁₅
4	0 0100	00i 10 0000 t ₁₆	00i 10 0000 t ₁₇	04e 00 0100 t ₁₈	00i 10 0000 t ₁₉
5	0 0101	00i 10 0000 t ₂₀	00i 10 0000 t ₂₁	00i 10 0000 t ₂₂	05e 00 0101 t ₂₃
6	0 0110	00i 10 0000 t ₂₄	00i 10 0000 t ₂₅	06e 00 0110 t ₂₆	00i 10 0000 t ₂₇
7	0 0111	00i 10 0000 t ₂₈	00i 10 0000 t ₂₉	00i 10 0000 t ₃₀	07e 00 0111 t ₃₁
8	0 1000	00i 10 0000 t ₃₂	00i 10 0000 t ₃₃	08e 00 1000 t ₃₄	00i 10 0000 t ₃₅
9	0 1001	00i 10 0000 t ₃₆	00i 10 0000 t ₃₇	00i 10 0000 t ₃₈	09e 00 1001 t ₃₉
10	0 1100	00i 10 0000 t ₄₀	00i 10 0000 t ₄₁	10e 00 1010 t ₄₂	00i 10 0000 t ₄₃
11	0 1011	00i 10 0000 t ₄₄	00i 10 0000 t ₄₅	00i 10 0000 t ₄₆	11e 00 1011 t ₄₇
12	0 1100	00i 10 0000 t ₄₈	00i 10 0000 t ₄₉	12e 00 1100 t ₅₀	00i 10 0000 t ₅₁
13	0 1101	00i 10 0000 t ₅₂	00i 10 0000 t ₅₃	00i 10 0000 t ₅₄	13e 00 1101 t ₅₅
14	0 1110	00i 10 0000 t ₅₆	00i 10 0000 t ₅₇	14e 00 1110 t ₅₈	00i 10 0000 t ₅₉
15	0 1111	00i 10 0000 t ₆₀	00i 10 0000 t ₆₁	00i 10 0000 t ₆₂	15e 00 1111 t ₆₃
16	1 0000	16e 01 0000 t ₆₄	00i 00 0000 t ₆₅	16i 00 0000 t ₆₆	00i 10 0000 t ₆₇
17	1 0001	16i 01 0000 t ₆₈	17e 01 0001 t ₆₉	00i 10 0000 t ₇₀	01i 00 0001 t ₇₁
18	1 0010	18e 01 0010 t ₇₂	17i 01 0001 t ₇₃	12e 01 0010 t ₇₄	19i 01 0011 t ₇₅
19	1 0011	02i 00 0010 t ₇₆	19e 01 0011 t ₇₇	00i 10 0000 t ₇₈	19i 00 0011 t ₇₉
20	1 0100	18i 01 0010 t ₈₀	20i 01 0100 t ₈₁	22i 01 0110 t ₈₂	20e 01 0100 t ₈₃
21	1 0101	04i 00 0100 t ₈₄	21e 01 0101 t ₈₅	05i 00 0101 t ₈₆	05i 00 0101 t ₈₇
22	1 0110	22e 01 0110 t ₈₈	21i 01 0101 t ₈₉	22e 01 0110 t ₉₀	23e 01 0111 t ₉₁
23	1 0111	06i 00 0110 t ₉₂	23e 01 0111 t ₉₃	00i 10 0000 t ₉₄	07i 00 0111 t ₉₅
24	1 1000	24e 01 1000 t ₉₆	20i 01 0100 t ₉₇	24e 01 1000 t ₉₈	28i 01 1100 t ₉₉
25	1 1001	08i 00 1000 t ₁₀₀	25e 01 1001 t ₁₀₁	00i 10 0000 t ₁₀₂	09i 00 1001 t ₁₀₃
26	1 1010	26e 01 1010 t ₁₀₄	25i 01 1001 t ₁₀₅	26e 01 1010 t ₁₀₆	27i 01 1011 t ₁₀₇
27	1 1011	10i 00 1010 t ₁₀₈	27e 01 1011 t ₁₀₉	00i 10 0000 t ₁₁₀	27i 00 1011 t ₁₁₁
28	1 1100	26i 01 1010 t ₁₁₂	28e 01 1100 t ₁₁₃	30i 01 1110 t ₁₁₄	28e 01 1100 t ₁₁₅
29	1 1101	12i 00 1100 t ₁₁₆	29e 01 1101 t ₁₁₇	00i 10 0000 t ₁₁₈	13i 00 1101 t ₁₁₉
30	1 1110	30e 01 1110 t ₁₂₀	29i 01 1101 t ₁₂₁	30e 01 1110 t ₁₂₂	31i 01 1111 t ₁₂₃
31	1 1111	14i 00 1110 t ₁₂₄	31e 01 1111 t ₁₂₅	00i 10 0000 t ₁₂₆	15i 00 1111 t ₁₂₇

Tabela 3: Matriz de Projeto.

A Tabela 4 busca esclarecer o processo de preenchimento da matriz de projeto, utilizando o mesmo exemplo fornecido para o conversor A/D, ou seja, uma tensão de entrada de 3,7V, com uma comparação inicial realizada com 8V.

Endereço em hexadecimal	Realimentações					Entradas		Saídas						Época matemática
	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀	X	X ₀	Q ₅	Q ₄	Q ₃	Q ₂	Q ₁	Q ₀	
00	0	0	0	0	0	0	0	0	1	1	0	0	0	t ₀
60	1	1	0	0	0	0	0	0	1	1	0	0	0	t ₉₆
61	1	1	0	0	0	0	1	0	1	0	1	0	0	t ₉₇
51	1	0	1	0	0	0	1	0	1	0	1	0	0	t ₈₁
50	1	0	1	0	0	0	0	0	1	0	0	1	0	t ₈₀
4A	1	0	0	1	0	1	0	0	1	0	0	1	0	t ₇₄
4B	1	0	0	1	0	1	1	0	1	0	0	1	1	t ₇₅
4F	1	0	0	1	1	1	1	0	0	0	0	1	1	t ₇₉
0F	0	0	0	1	1	1	1	0	0	0	0	1	1	t ₁₅

Tabela 4: Exemplo de geração da Matriz de Projeto.

Nota-se que o endereço é fornecido pelas variáveis Y₄ Y₃ Y₂ Y₁ Y₀ X X₀. Por exemplo: t₈₀ corresponde em binário a 1010000 e a 50 em hexadecimal. A época matemática t₀ representa a condição inicial do sistema, um *reset* que zera todas as entradas, endereçando desta forma o valor inicial para a comparação (endereço 00h com dado 18h).

O comparador confronta a tensão de entrada Vin com a tensão oriunda do conversor digital-para-analógico V_{DA}, levando a saída do comparador para nível alto ou baixo, conforme a situação.

A saída do comparador é uma entrada X na memória, juntamente com X₀, que é o passo temporal, e com as variáveis de realimentação. As entradas da memória são na verdade seu endereçamento. Observa-se, então, que as realimentações e as variáveis de entrada X e X₀ definem a cada passo temporal um novo endereçamento e, conseqüentemente, um novo dado de saída.

4.3. Tabela Verdade Equivalente

A tabela verdade é uma etapa que tem a finalidade de organizar de forma coerente os dados que foram obtidos através da matriz de projeto, a fim de preencher o mapa de gravação da memória. Uma alternativa tecnológica de baixo custo é a conhecida Eprom. Para a tabela verdade são consideradas como variáveis de entrada Y₄ Y₃ Y₂ Y₁ Y₀ X X₀, sendo X₀ o bit menos significativo. Com 7 variáveis são possíveis 2⁷=128 combinações (linhas), o que resulta em uma tabela muito extensa e que não é exibida. As variáveis de saída são Q₅ Q₄ Q₃ Q₂ Q₁ Q₀, sendo Q₀ o bit menos significativo.

Não é essencial a construção da tabela verdade equivalente, já que é possível preencher o mapa de gravação da memória a partir da matriz de projeto. Basta, para tanto, converter os dados que estão em binário para hexadecimal e tomar a posição correta que é indicada através da época matemática (t) e seu respectivo índice. A capacidade da memória é determinada avaliando-se o número de entradas e saídas do sistema. Neste caso, uma Eprom 2716 pode ser utilizada, pois possui 11 entradas e 8 saídas.

4.4. Mapa de Gravação da Memória

O mapa final que será gravado na memória Eprom é visto na Tabela 5. O endereço é formado por CENTENA DEZENA UNIDADE, como exemplificado por $t_{94}=05Eh$. O traçado pontilhado indica o caminho percorrido para a tensão de 3,7V, dada como exemplo anteriormente.

		CENTENA DEZENA									
		00	01	02	03	04	05	06	07		
UNIDADE	0	18 t_0	20 t_6	20 t_{32}	20 t_{48}	10 t_4	12 t_0	18 t_{96}	1A t_{112}		
	1	00 t_1	20 t_7	20 t_{33}	20 t_{49}	00 t_5	14 t_1	14 t_{97}	1C t_{113}		
	2	00 t_2	04 t_8	08 t_{34}	0C t_{50}	00 t_6	16 t_2	18 t_{98}	1E t_{114}		
	3	20 t_3	20 t_9	20 t_{35}	20 t_{51}	20 t_7	20 t_3	1C t_{99}	1C t_{115}		
	4	20 t_4	20 t_0	20 t_{36}	20 t_{52}	10 t_8	04 t_4	08 t_{100}	0C t_{116}		
	5	20 t_5	20 t_1	20 t_{37}	20 t_{53}	11 t_9	15 t_5	19 t_{101}	1D t_{117}		
	6	20 t_6	05 t_2	20 t_{38}	20 t_{54}	20 t_0	05 t_6	00 t_{102}	00 t_{118}		
	7	01 t_7	00 t_3	09 t_{39}	0D t_{55}	01 t_1	00 t_7	09 t_{103}	0D t_{119}		
	8	20 t_8	20 t_4	20 t_{40}	20 t_{56}	12 t_2	16 t_8	1A t_{104}	1E t_{120}		
	9	20 t_9	20 t_5	20 t_{41}	20 t_{57}	11 t_3	15 t_9	19 t_{105}	1D t_{121}		
	A	02 t_0	06 t_6	0A t_{42}	0E t_{58}	12 t_4	16 t_0	1A t_{106}	1E t_{122}		
	B	20 t_1	20 t_7	20 t_{43}	20 t_{59}	13 t_5	17 t_1	1B t_{107}	1F t_{123}		
	C	20 t_2	20 t_8	20 t_{44}	20 t_{60}	02 t_6	06 t_2	0A t_{108}	0F t_{124}		
	D	20 t_3	20 t_9	20 t_{45}	20 t_{61}	13 t_7	17 t_3	1B t_{109}	1F t_{125}		
	E	20 t_4	20 t_0	20 t_{46}	20 t_{62}	20 t_8	20 t_4	20 t_{110}	20 t_{126}		
	F	03 t_5	07 t_1	0B t_{47}	0F t_{63}	03 t_9	07 t_5	0B t_{111}	0F t_{127}		

Tabela 5: Mapa de Gravação da Memória EPROM.

4.5. Diagrama de Blocos do Projeto

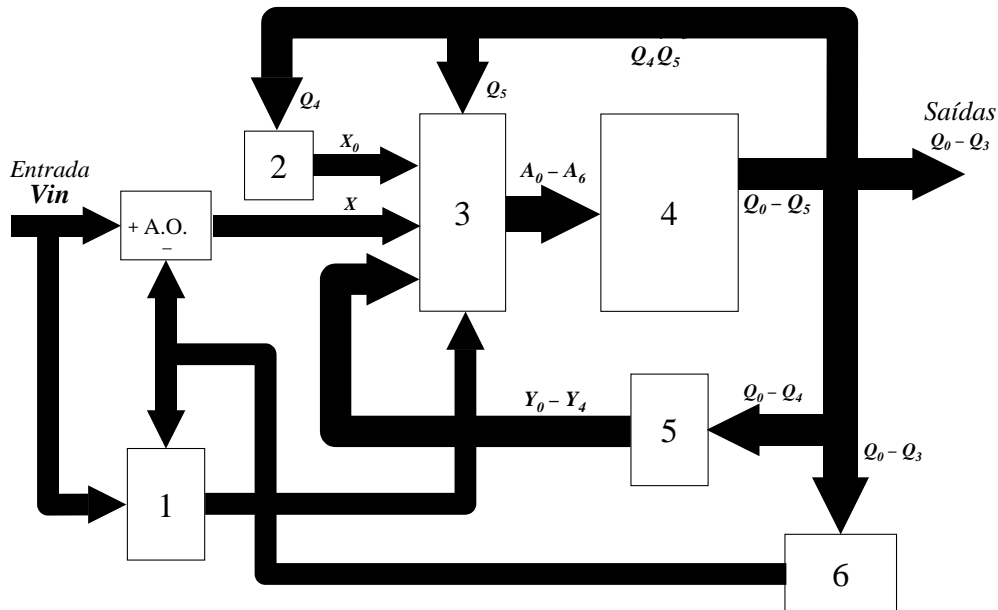


Figura 3: Diagrama de Blocos.

O diagrama de blocos (Figura 3) representa a finalização do projeto. As descrições dos blocos são dadas a seguir.

a) Bloco 1 - Detector de mudança

Sua função é verificar se a tensão de entrada, após ter seu valor identificado, não variou para um valor maior do que V_{in} . Após o término das comparações, a saída do comparador (amplificador operacional) permanece em nível alto, indicando que V_{in} é maior que V_{DA} (bloco 6), como no exemplo dos 3,7V. Porém, se a tensão de entrada V_{in} aumentar, o comparador continuará com nível alto na saída, indicando que "não houve" nenhuma variação na tensão a ser identificada. O bloco detector de mudança identifica essa alteração na tensão de entrada V_{in} e reseta o sistema para que possa iniciar uma nova leitura.

b) Bloco 2 - Passo Temporal

Possibilita o controle síncrono ou assíncrono, dependendo da implementação.

c) Bloco 3 - Reset

É responsável pela geração de zero nas entradas e conseqüente endereçamento, levando o primeiro dado para a comparação de V_{in} .

d) Bloco 4 - Memória

É a memória Eprom definida. Nela estão armazenados os dados que são utilizados nas comparações.

e) Bloco 5 - Realimentação

As linhas de realimentação enviam uma parte dos novos dados para as vias de endereçamento, apontando um novo endereço. O novo endereço apontado na saída pela realimentação será novamente um estado interno estável.

f) Bloco 6 - Conversor Digital-para-Analógico (D/A)

Converte o dado recebido da memória em um sinal analógico para as comparações com a tensão de entrada V_{in} .

g) Variáveis de saída

Q_4 → Além de ser uma variável de realimentação, é também utilizada para desabilitar o passo temporal (bloco 2) e habilitar o detector de mudança (bloco 1) ao término de cada comparação. Desabilita com nível zero.

Q_5 → Serve para *resetar* as entradas cada vez que por um motivo indeterminado o sistema for para um endereço que não faz parte do processo. Evita, desta forma, que um valor de V_{in} seja convertido erroneamente.

4.6. Circuito Implementado

A Figura 4 apresenta o circuito implementado com componentes eletrônicos comerciais.

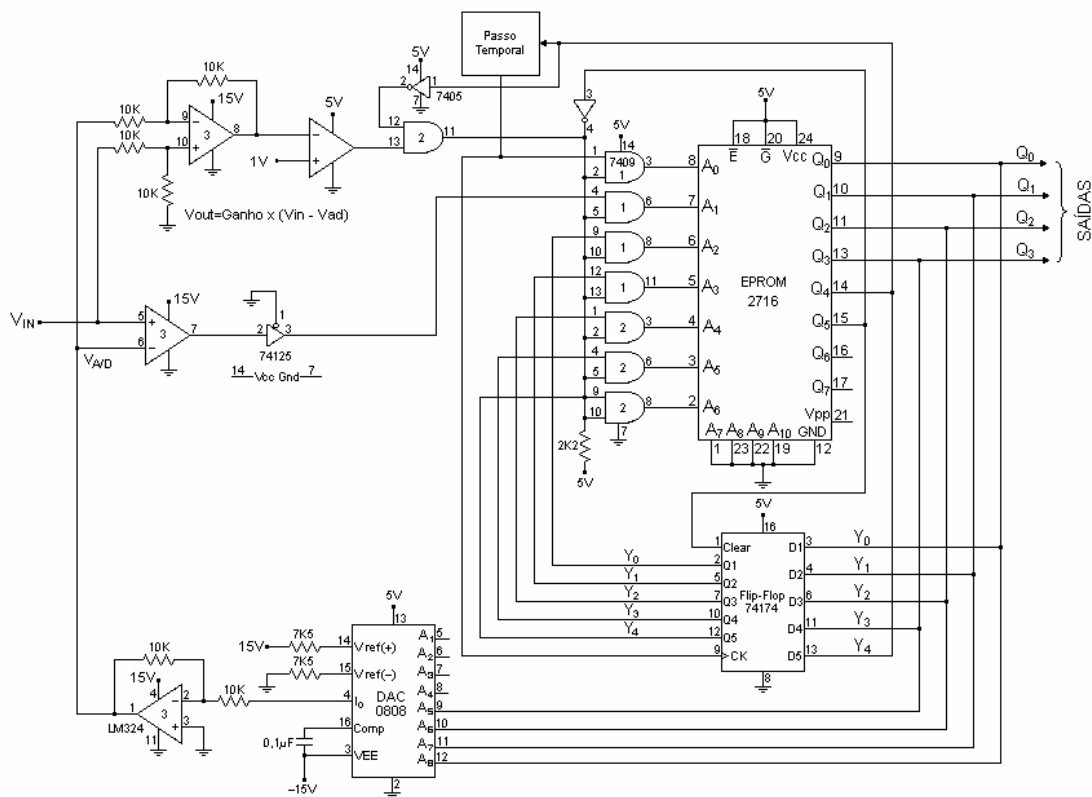


Figura 4: Circuito Implementado.

5. Conclusão

As soluções que adotam alguma implementação baseada em arquitetura computacional assíncrona tendem a ser cada vez mais efetivas, baratas e de fácil implementação. Na verdade, uma solução como esta apresentada mostra claramente sua viabilidade tecnológica e uma constante melhoria pode vir a justificar a aplicação em casos que mesclam circuitos síncronos e assíncronos, com sub-circuitos dedicados e integrados [WEB3].

6. Bibliografia

[CAMI] CAMILO, Daniel; YABU-UTI, João Batista T.; YANO, Yuzo. **Circuitos Lógicos**. São Paulo: Livraria Ciência e Tecnologia, 1984.

[CUES] CUESTA, L.; PADILLA, A. Gil; REMIRO, F. **Electrónica Digital**. Lisboa: Mc Graw Hill, 1994.

[DAVI] DAVIS, Al; NOWICK, Steven M. **An Introduction to Asynchronous Circuit Design**. Salt Lake City, University of Utah; New York, Columbia University, September 19th, 1997.

[DELP] DEL PICCHIA, Walter. **Métodos Numéricos Para Resolução de Problemas Lógicos**. São Paulo: Edgard Blücher, 1993.

[EBER] EBERGEN, Jo. **Circuits Without Clocks: What Makes Them Tick?** Mountain View, Sun Microsystems Laboratories, May 13th, 1998.

[ERCE] ERCEGOVAC, Milos; LANG, Tomás; MORENO, Jaime H. **Introdução aos Sistemas Digitais**. Porto Alegre: Bookman, 2000.

[MART] MARTINS, W. Waneck. **Esção (n-m-p): Um Computador não-Von Neumann**. São Paulo: Cartgraf, 1985.

[MYER] MYERS, Chris. **Asynchronous Circuit Design**. New York: John Wiley & Sons, 2001.

[NOWI] VAN BERKEL, C. H.; JOSEPHS, Mark B.; NOWICK, Steven M. Scanning the Technology: Applications of Asynchronous Circuits. **Proceedings of the IEEE**, v.87, n.2, p.223-233, Feb. 1999.

[PESO] PESSOTA, Roberto C. **C.L.P. “Não-Von” a Tempo Real, Matematicamente Programável**. São Paulo: Tese de Doutorado, EPUSP, 1993.

[PHIS] PHISTER Jr., M. **Logical Design of Digital Computers**. New York: John Wiley & Sons, 1958.

[SUTH] SUTHERLAND, Ivan E.; LEXAU, Jon K. Designing Fast Asynchronous Circuits. **ASYNC 2001 Seventh International Symposium on Advanced Research in Asynchronous Circuits and Systems**, Salt Lake City, Proceedings, March 11-14, 2001.

[TANE] TANENBAUM, Andrew S. **Organização Estruturada de Computadores**. Rio de Janeiro: Livros Técnicos e Científicos, 1999.

[TIND] TINDER, Richard F. **Engineering Digital Design**. 2.ed. San Diego: Academic Press, 2000.

[TOCI] TOCCI, Ronald J. **Sistemas Digitais**. Rio de Janeiro: Prentice Hall do Brasil, 1994.

[WEB1] WEBER, Leo; FIGUEREDO, Melissa G.; KLEIN, Pedro A. T. A transformada numérica. **Logos**, Canoas, v. 11, p.45-49, dez. 1998.

[WEB2] WEBER, Leo; KOPLIN, Gustavo R.; FIGUEREDO, Melissa G.; DAPPER, Roque E. Uma Arquitetura Computacional Alternativa. **Saber Eletrônica**, São Paulo, n. 336, p.37-42, jan. 2001.

[WEB3] WEBER, Leo. Porque um circuito computacional pode trabalhar sem relógio? **Revista Liberato**, Novo Hamburgo, v.2, n.2, p.39-43, nov. 2001.

[WEB4] WEBER, Leo. Aplicação da transformada numérica na resolução de problemas da álgebra booleana. **Revista Liberato**, Novo Hamburgo, v.3, n.3, p.83-91, nov. 2002.