

Documentação de construção de relógio digital

Eduardo Almeida Cavalcanti de Melo
 Higor Gabriel de Freitas
 Hugo Cardoso Ferreira de Araújo
 Kauã Lima de Souza

¹Escola de Artes, Ciências e Humanidades, EACH-USP

Resumo

Este relatório consiste na descrição das decisões e simplificações tomadas na construção de um relógio digital usando flip-flops, portas lógicas, oscilador e display para esse feito. A tarefa consistiu na construção de um chip contador (usando um total de 17 flip-flops JK e 3 flip-flop D); um separador de binário-decimal; um decodificador dos dígitos binários para o display.

1 Introdução

A tarefa assumida consistiu em construir um relógio capaz de mostrar o horário usando displays de 7 segmentos, no formato hora:minuto:segundo. Para tanto, foi necessário mapear um total de 6 displays para a exibição correta do horário no formato. Fomos restritos a usar somente flip-flops (especialmente JK), portas lógicas (como OR, AND, etc), display de 7 segmentos e um oscilador para gerar pulsos.

Dividimos o trabalho em três partes:

1. **Construção de três contadores síncronos**, sendo dois contadores de 0 a 59 (exibindo um total de 6 flip-flops JK e 1 flip-flop D em cada um), e outro de 0 a 23 (exibindo um total de 5 flip-flops JK com parada e 1 flip-flop D).
2. **Construção de separador binário-decimal**, que foi utilizado para separar a saída binária dos contadores no dígito de sua dezena e no dígito de sua unidade.
3. **Decodificador dos dígitos do display**, que utiliza os dígitos da saída do separador para transformar no padrão de exibição dos displays.

2 Construção dos contadores

2.1 Contadores de 0 a 59

Os contadores de 0 a 59 foram feitos usando 6 flip-flops JK ($S_1, S_2, S_3, S_4, S_5, S_6$) com borda de descida. Ou seja, construímos a priori um contador

crescente de 0 a 63 e depois adicionamos um critério de parada em 60, uma vez que:

$$(59)_{10} = 111011b$$

$$(60)_{10} = 111100b$$

basta verificar se os 4 dígitos mais significativos das saídas dos flip-flops são 1, i.e, para qualquer valor > 60 temos (chamando a saída Q dos flip-flops de $Q_1, Q_2, Q_3, Q_4, Q_5, Q_6$ do bit menos significativo para o mais significativo):

$$Q_6 \cap Q_5 \cap Q_4 \cap Q_3$$

Fizemos essa verificação por meio de portas AND nas saídas dos flip-flops, que leva ao *CLEAR* de todos os flip-flops, resetando o valor para qualquer valor ≥ 60 .

2.2 Contador de 0 a 23

Os contadores de 0 a 23 foram feitos usando 5 flip-flops JK com borda de descida. De modo análogo, construímos a priori um contador crescente de 0 a 31 e depois adicionamos um critério de parada em 24, uma vez que:

$$(23)_{10} = 10111b$$

$$(24)_{10} = 11000b$$

do mesmo modo, para detectar qualquer número superior ou igual a 24, basta verificar se os 2 dígitos mais significativos dos flip-flops são 1:

$$Q_5 \cap Q_4$$

E do mesmo modo, fazemos essa verificação por AND levando ao *CLEAR* dos flip-flops em qualquer valor ≥ 24 .

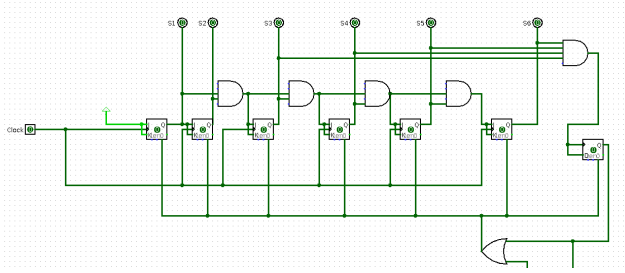


Figura 1: Contador de 0 a 59 com parada.

2.3 Construção do sistema de CLEAR

Uma observação importante é que como a detecção do critério de parada é baseado na saída síncrona do flip-flop, isso leva a um atraso de meio clock em qualquer situação de *CLEAR* no contador e na passagem do clock para o contador seguinte. Esse problema pode ser contornado de dois modos:

1. **Realizar um *CLEAR* assíncrono:** o que leva ao problema de limpar o valor do contador durante os estados intermediários (i.e, como não há sincronia de saída dos flip-flops, na hipótese de um estado temporário de mudança dos valores dos flip-flops posteriores, poderia ocorrer um *CLEAR* inadequado;
2. **Utilizar um flip-flop D com borda de subida:** essa foi a solução adotada na construção do exercício-programa, que consiste em fazer um flip-flop D com a borda inversa às utilizadas no flip-flop JK com ativação perpétua (ligando a saída do sistema de detecção tanto em sua entrada de clock e em sua entrada D).

Por essa razão, a saída com base no critério de parada é passada para um flip-flop D em cada contador, totalizando três flip-flops D. Outra observação importante é que o flip-flop D tem seu *CLEAR* com base em sua própria saída, o que garante que ao limpar as saídas do contador ele tenha seu valor zerado.

O circuito conta também com um botão de *RESET* associado a todos os flip-flops e ao sistema de *CLEAR* para facilmente iniciar um *CLEAR* em todos os flip-flops.

3 Construção do separador binário-decimal

As saídas dos contadores são binárias e vão ou de 0 a 59 ou de 0 a 23, o que não podem ser diretamente colocadas nos displays de 7 segmentos, que têm as demonstrações de valores segundo a seguinte tabela:

Saída Decodificador							Valor Display
a	b	c	d	e	f	g	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

O que demonstra que devemos separar o dígito da dezena e o da unidade da saída dos contadores, levá-los a um decodificador do display, para enfim mostrar seu valor.

Para tanto, construímos um separador binário-decimal que separa o dígito decimal da dezena e da unidade em binário (e.g, $59 = 111011b$, devemos separar em $5 = 000101b$ e $9 = 001001b$) construímos a tabela de equivalência dos números binários em expressões booleanas:

Nº	A	B	C	D	E	F	Representação
0	0	0	0	0	0	0	$\overline{A}\overline{B}\overline{C}\overline{D}\overline{E}\overline{F}$
1	0	0	0	0	0	1	$\overline{A}\overline{B}\overline{C}\overline{D}\overline{E}F$
2	0	0	0	0	1	0	$\overline{A}\overline{B}\overline{C}\overline{D}E\overline{F}$
3	0	0	0	0	1	1	$\overline{A}\overline{B}\overline{C}\overline{D}EF$
4	0	0	0	1	0	0	$\overline{A}\overline{B}\overline{C}D\overline{E}\overline{F}$
...

Essa tabela permite montar a simplificação em soma de produtos e mapas de Karnaugh para o dígito da dezena (4 possíveis) e os da unidade (4 possíveis) nos dígitos equivalentes. Obtemos as expressões ao somar o valor das colunas para obter as expressões booleanas no formato de simplificação.

A tabela para as simplificações foram colocadas no arquivo "*Tabela de Simplificações do Separador Binário-Decimal e Display.xlsx*". As simplificações das somas de produto foram feitas em caderno, assim como os mapas de Karnaugh e estão

Digito(4, dezenas)	Digito(3, dezenas)	Digito(2, dezenas)	Digito(1, dezenas)
0	AB'C	AB'C'	A'B'CD
	ABD'	A'BD	ABC'D
	ABC'	A'BC	AB'C'
			A'BC'D'
Digito(4, unidade)	Digito(3, unidade)	Digito(2, unidade)	Digito(1, unidade)
A'BC'D'E	A'B'C'D	AB'CE	ABCD'
A'BCDE'	A'B'CDE	A'B'C'E	A'B'CD'E
AB'C'DE	A'BC'D'E'	A'C'DE	ABC'D'E
ABC'D'E'	A'BCD'	A'BCD'E	
ABCD'E	AB'C'D'E	AB'C'E'	Digito(1, unidade)
	AB'CD	A'B'CDE'	F
	ABC'DE	A'BC'D'E'	
	ABCD'E'	ABCD'E'	

Figura 2: Resultados para cada dígito do separador binário-decimal.

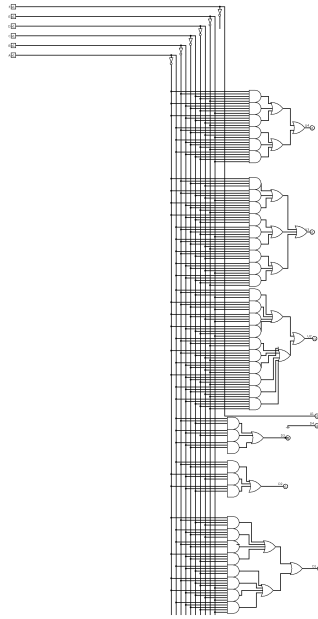


Figura 3: Circuito Separador Binário-Decimal

no diretório "*Simplificações e MK*" em seis arquivos .jpeg.

Observação: Para o dígito 4 da dezena e dígito 1 da unidade temos as seguintes simplificações (que são simples o suficiente para não precisar ser feitas em caderno):

$$Digito(4, d) = 0 + 0 + \dots + 0 = 0$$

$$Digito(1, u) = F + F + \dots + F = F$$

A partir desses resultados, conectamos a saída dos contadores que chega em um binário de 6 bits, na entrada do separador binário-decimal e mapeamos usando portas *AND* e *OR* com base nas simplificações das tabelas obtidas.

D1	AB\CD	00	01	11	10		CA'
	00	1	0	1	1		C'AB'
	01	0	1	1	1		C'D'B'
	11	0	0	0	0		DA'B
	10	1	1	0	0		
D2	AB\CD	00	01	11	10		A'B'
	00	1	1	1	1		C'B'
	01	1	0	1	0		C'D'A'
	11	0	0	0	0		CDA'
	10	1	1	0	0		
D3	AB\CD	00	01	11	10		A'B
	00	1	1	1	0		C'A'
	01	1	1	1	1		C'B'
	11	0	0	0	0		DA'
	10	1	1	0	0		
D4	AB\CD	00	01	11	10		C'AB'
	00	1	0	1	1		CA'B'
	01	0	1	0	1		CD'A'
	11	0	0	0	0		C'D'B'
	10	1	1	0	0		C'DA'B

Figura 4: Mapa de Karnaugh do Decodificador do Display

4 Construção do Decodificador dos Dígitos do Display

A partir das saídas para cada dígito obtidas no separador binário-decimal, agora basta mapear os valores da tabela do display para mostrar cada valor entre 0 a 9.

Display	D1	D2	D3	D4	D5	D6	D7
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
...							

Tabela 1: Tabela de segmentos do display (D1 a D7)

Usando a soma de produtos em mapas de Karnaugh, conseguimos simplificar as expressões obtidas. A tabela inteira se encontra na seção "*Display*" no arquivo "*Tabela de Simplificações do Separador Binário-Decimal e Display.xlsx*".

De modo análogo ao separador decimal, montamos o chip decodificador do display utilizando portas *OR* e *AND*.

5 União dos chips criados para montagem do relógio

A interligação dos chips criados permitiu a criação do relógio digital capaz de fazer contagem no formato hora:minuto:segundo.

Um detalhe importante é que, devido à ausência do *chip 555* no programa Logisim, o clock deve ser manualmente colocado a cada simulação em 2 hz para medição de tempo em tempo real. Além disso, a entrada do separador decimal mais inferior em sua entrada A foi aterrada pois sempre possui valor zero.

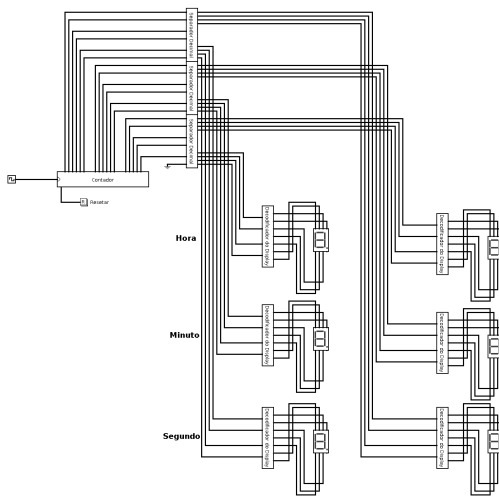


Figura 5: Relógio digital completo.

O circuito final funcionou bem e reforçou os principais conceitos vistos em aula e é capaz de mostrar as 24 horas completas, resetando ao final da vigésima quarta hora.

O projeto do relógio digital ajudou a entender na prática circuitos síncronos, lógica combinacional e sistemas digitais. A montagem dos contadores, do separador binário-decimal e do decodificador fez relevante a importância da organização em blocos. Mesmo com limitações do Logisim, como a falta do 555, o uso de um clock manual permitiu testar toda a lógica.