Segunda Prova de Circuitos Digitais – Q3 2013 – Turma NABC1517SA – 31/jan/2014

Nome: _____ RA: _____

Resumão de Circuitos Digitais

(versão 3, gerada em 2014-01-30 16:23)

ATENÇÃO: este texto **deve** ser retornado ao professor junto com a prova.

Não traga nenhum material de consulta para a prova. Este resumo será distribuído pelo professor junto com o caderno de prova e será a única fonte de consulta permitida durante o exame.

Os espaços em branco podem ser usados para rascunho.

1 Introdução

Bases de numeração: A notação $A = (a_{n-1}a_{n-2}\ldots a_1a_0)_2$ indica que o número inteiro A é representado na base 2 pelos n algarismos binários a_{n-1} (algarismo mais significativo), a_{n-2} (segundo algarismo mais significativo), $\ldots$, a_1 (segundo algarismo menos significativo), a_0 (algarismo menos significativo). O valor de A é dado pela expressão $a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \ldots + a_1 \times 2^1 + a_0 \times 2^0$.

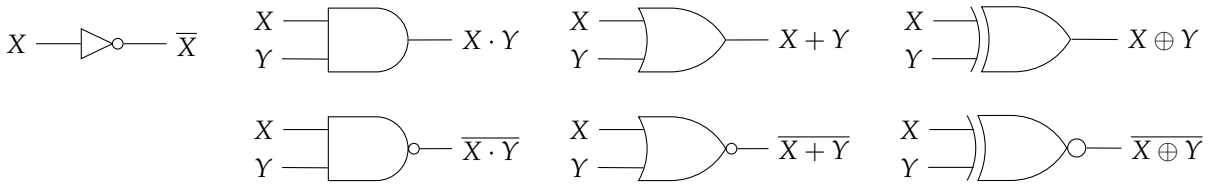
Conversão de números para a base 2: Para números inteiros, use o algoritmo das divisões sucessivas por 2. Para números fracionários entre 0 e 1, use o algoritmo das multiplicações sucessivas por 2.

Palavras de dados e formatos de números: considere um inteiro representado em uma palavra de dados com n bits $\boxed{a_{n-1}}\boxed{a_{n-2}}\ldots\boxed{a_1}\boxed{a_0}$. Dependendo do formato adotado, o seu valor será...

formato	valor
inteiro sem sinal	$(a_{n-1}a_{n-2}\ldots a_1a_0)_2$
sinal-magnitude	se $a_{n-1} = 0$, $+(a_{n-2}\ldots a_1a_0)_2$
	se $a_{n-1} = 1$, $-(a_{n-2}\ldots a_1a_0)_2$
complemento de dois	se $a_{n-1} = 0$, $+(a_{n-2}\ldots a_1a_0)_2$
	se $a_{n-1} = 1$, $-[(\overline{a_{n-2}}\ldots\overline{a_1}\,\overline{a_0})_2 + 1]$

Operações lógicas: negação (not) $\overline{X}$ conjunção (and) $X \cdot Y$ disjunção (or) $X + Y$
disjunção exclusiva (xor) $X \oplus Y$
Obs.: $X \oplus Y = \overline{X}Y + X\overline{Y}$

Portas lógicas: circuitos que executam operações lógicas.

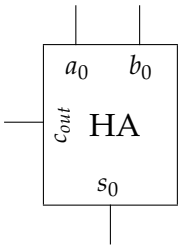


2 Síntese de circuitos digitais combinacionais

0. Entenda o problema e obtenha a tabela verdade para cada saída.
1. Obtenha e simplifique a expressão lógica para cada saída. Para até 4 variáveis (ou até 6 para os muito atentos), é possível simplificar manualmente usando mapas de Karnaugh.
2. Desenhe o diagrama do circuito.
3. Analise o circuito e verifique as saídas.
4. Simulação e montagem.

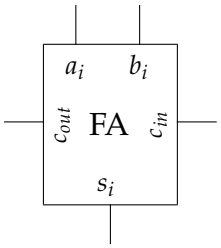
3 Blocos lógicos básicos

Meio-somador (half-adder)



- Entradas: a_0, b_0, dois bits a serem somados.
- Saídas: s_0, a soma aritmética das entradas;
  c_{out}, vai-um.

Somador completo (full-adder)

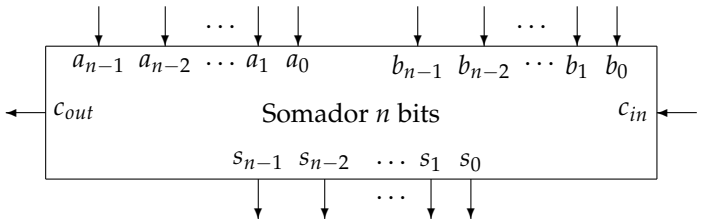


- Entradas: a_i, b_i, c_{in}, três bits a serem somados.
- Saídas: s_i, soma aritmética das entradas;
  c_{out}, vai-um.

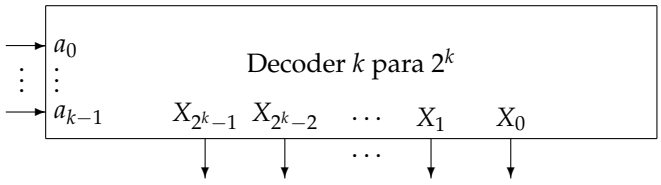
Somador completo de n bits

- Entradas: $a_{n-1}, a_{n-2}, \ldots, a_1, a_0$, bits que representam um inteiro A;
  $b_{n-1}, b_{n-2}, \ldots, b_1, b_0$, bits que representam um inteiro B;
  c_{in}, vem-um.
- Saídas: $s_{n-1}, s_{n-2}, \ldots, s_1, s_0$, representando o resultado da soma aritmética $A + B + c_{in}$;
  c_{out}, vai-um.

Os inteiros A e B ambos devem estar no formato inteiro sem sinal ou no formato complemento de dois. A saída estará no mesmo formato da entrada.

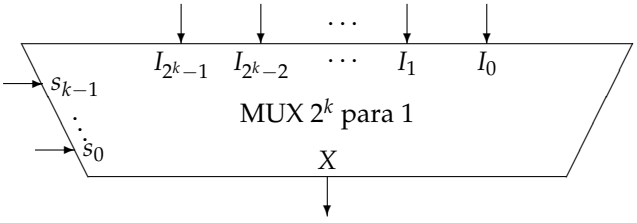


Decodificador (decoder) k para 2^k – Entradas: $a_{k-1}, a_{k-2}, \ldots, a_1, a_0$, as entradas de endereço. Saídas: $X_0, X_1, \ldots, X_{2^k-1}$, onde cada combinação de bits de endereço ativa apenas uma saída X_j tal que $j = (a_{k-1}a_{k-2}\ldots a_1a_0)_2$.



Codificador (encoder) 2^k para k: circuito que faz a operação inversa do codificador. Entradas $X_0, X_1, \ldots, X_{2^k-1}$ e saídas $a_{k-1}, a_{k-2}, \ldots, a_1, a_0$.

Multiplexador (mux) 2^k para 1 – Entradas: $I_0, I_1, \ldots, I_{2^k-1}$, as entradas de dados; $s_{k-1}, s_{k-2}, \ldots, s_1, s_0$, as entradas de seleção. Saída: X, onde X tem o valor da entrada I_j tal que $j = (s_{k-1}s_{k-2}\ldots s_1s_0)_2$. O mux funciona como uma chave seletora.



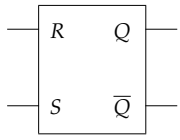
Demultiplexador (demux) 1 para 2^k: circuito que faz a operação inversa do multiplexador. Entradas: X (entrada de dado) e $s_{k-1}, \ldots, s_0$ (entradas de seleção). Saídas: $I_0, I_1, \ldots, I_{2^k-1}$.

4 Circuitos digitais sequenciais

Notação: Q_i = estado atual da variável Q; Q_{i-1} = estado anterior da variável Q; $\times$ = *don't care*

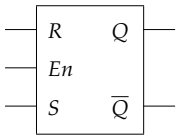
Latches: circuitos digitais sequenciais básicos, que guardam informação sobre o estado anterior. Cada um dos latches abaixo possui uma saída Q, e uma saída com estado inverso $\overline{Q}$.

Latch R-S (Reset-Set)



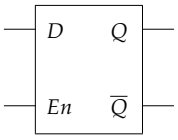
R	S	Q_i	
0	0	Q_{i-1}	(mantém)
0	1	1	(set)
1	0	0	(reset)
1	1	entrada proibida	

Latch R-S c/ Enable



Quando $En = 0$, as saídas não se alteram. Quando $En = 1$, funciona como o latch R-S comum.

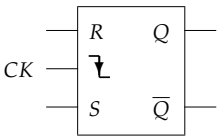
Latch D (Data)



En	D	Q_i	
0	$\times$	Q_{i-1}	(mantém)
1	0	0	(reset)
1	1	1	(set)

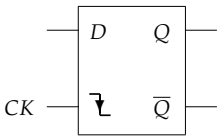
Flip-flops: latches sensíveis a uma das bordas do *clock*. O símbolo ⊓ denota sensibilidade à borda de descida (transição $1 \rightarrow 0$).

Flip-flop R-S



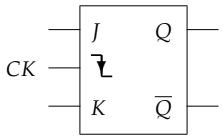
R	S	CK	Q_i	
0	0	$\times$	Q_{i-1}	(mantém)
0	1	$1\rightarrow0$	1	(set)
1	0	$1\rightarrow0$	0	(reset)
1	1	$1\rightarrow0$	entrada proibida	

Flip-flop D



D	CK	Q_i	
0	$1\rightarrow0$	0	(reset)
1	$1\rightarrow0$	1	(set)

Flip-flop J-K (Jump-Kill)



J	K	CK	Q_i	
0	0	$\times$	Q_{i-1}	(mantém)
0	1	$1\rightarrow0$	0	(kill)
1	0	$1\rightarrow0$	1	(jump)
1	1	$1\rightarrow0$	$\overline{Q_{i-1}}$	(inverte)

Entradas assíncronas: é possível alterar o circuito interno dos flip-flops e adicionar entradas assíncronas para *preset*, que coloca a saída Q em 1, e *clear*, que coloca a saída Q em 0. As entradas assíncronas afetam a saída Q imediatamente, não dependendo de nenhuma borda do clock.

Os sinais preset e clear podem ser ativos em nível *alto* ou em nível *baixo*:

Entradas assíncronas ativas em nível **baixo**

PRE	CLR	efeito no flip-flop
1	**0**	clear (zera Q assincronamente)
0	1	preset (coloca Q em 1 assincron.)
1	1	funcionamento normal do flip-flop
0	**0**	entrada probida

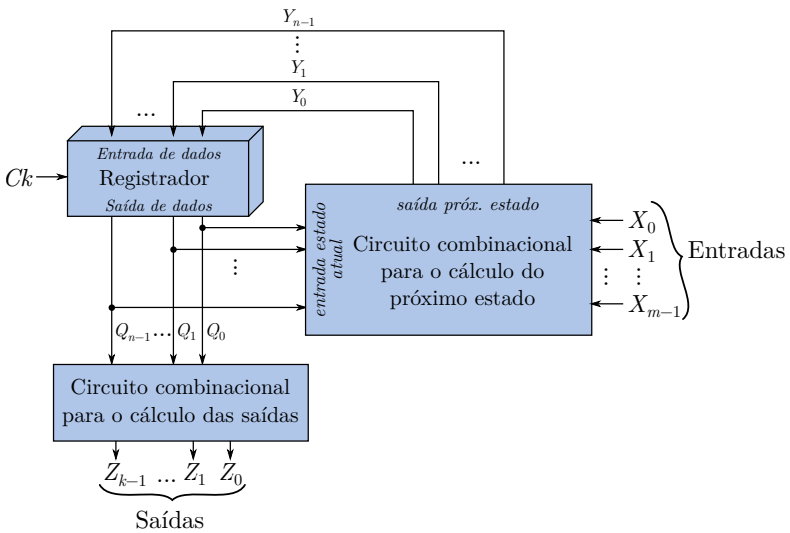
Entradas assíncronas ativas em nível **alto**

PRE	CLR	efeito no flip-flop
0	**1**	clear (zera Q assincronamente)
1	0	preset (coloca Q em 1 assincron.)
0	0	funcionamento normal do flip-flop
1	**1**	entrada proibida

5 Projeto de máquinas de estados

Para construir uma máquina de estados segundo o modelo ao lado:

0. Entenda o problema e obtenha o diagrama de estados.
1. Monte tabela de transição para próximo estado.
2. Determine e otimize expressões para saídas do próximo estado $Y_{n-1}, \ldots, Y_0$
3. Desenhe diagrama do circuito combinacional para cálculo do próximo estado
4. Determine e otimize expressões para saídas $Z_{k-1}, \ldots, Z_0$ em função de $Q_{n-1}, \ldots, Q_0$
5. Desenhe diagrama do circuito combinacional para cálculo das saídas
6. Desenhe diagrama unindo as três partes do circuito.



Esta página pode ser usada para rascunho. Por favor, não use a mesa como folha de rascunho!

Aguarde instrução para virar a prova!