

**USP - ICMC - SSC
SSC 0111 (Lab ELD I) - 2o. Semestre 2011**

**Disciplina de
Laboratório de Elementos de Lógica Digital I
SSC-0111**

Prof. Fernando Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Estagiário PAE: Diogo Ortiz Correa

Email: diogosoc [at] { icmc. usp. br }

Web: <http://www.icmc.usp.br/~fosorio/>

Wiki ICMC: [http://wiki.icmc.usp.br/index.php/SSC-111\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-111(fosorio))

Aula 10 – ULA e Circuitos Combinacionais

Agenda: (continuação)

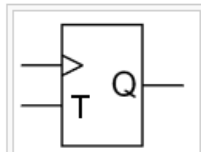
1. Somador / Subtrator Aula 09
Representação em Complemento de 2
Subtrator e Comparador ($< = >$)
2. Multiplexador / Demultiplexador (Mux/Demux)
Decodificador e Display de 7 segmentos
3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D) Aula 10
4. Contador (+1)
5. Shift Register: Shift Left ($\times 2$), Shift Right ($\div 2$)

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

Flip-flop T (Troca)



O símbolo do flip-flop T, onde > é a entrada de clock, T é a entrada de toggle e Q é informação de saída armazenada.

$$Q_{seguinte} = T \oplus Q$$

e pela tabela verdade:

T	Q	Q*
0	0	0
0	1	1
1	0	1
1	1	0

Memórias de 1 bit

Flip-Flop T

Altera estado da memória (Toggle)

Armazena 1 bit

Sinal T => Troca o estado armazenado

Q* → Estado seguinte do Q

3

<http://pt.wikipedia.org/wiki/Flip-flop>

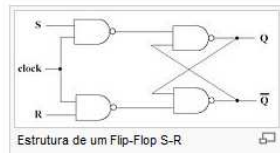
Agosto 2008

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

Flip-flop S-R Sincrono



Estrutura de um Flip-Flop S-R

Memórias de 1 bit

Flip-Flop SR ou RS (Set – Reset)

Armazena 1 bit

Sinal S => Set (Armazena 1)

Sinal R => Reset (Armazena 0)

Q	Q*	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Sugestão de Material de Consulta:

Wikipedia

<http://pt.wikipedia.org/wiki/Flip-flop>

[http://en.wikipedia.org/wiki/Flip-flop_\(electronics\)](http://en.wikipedia.org/wiki/Flip-flop_(electronics))

e Play-Hookey

www.play-hookey.com

http://www.play-hookey.com/digital/rs_nand_flip-flop.html

Simulador RS, JK <http://www.play-hookey.com/digital>

4

Q* → Estado seguinte do Q

<http://pt.wikipedia.org/wiki/Flip-flop>

Agosto 2008

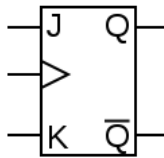
3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

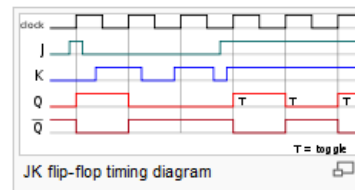
Quartus: Sugestão uso do JK
 Blocos pré-definidos...
 Primitivos – Storage - jkff

Q	J	K	Q[t+1]
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)
 Armazena 1 bit
 Sinal J => Set (Armazena 1)
 Sinal K => Reset (Armazena 0)
 Sinal J e K simultâneos => Inverte



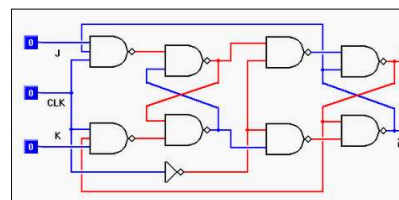
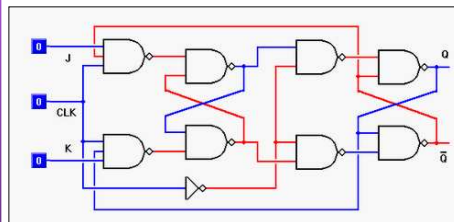
3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)



Simulador: Site Play-Hookey

Material de Consulta Complementar:
 Play-Hookey Simulador RS, JK
<http://www.play-hookey.com/digital>

Flip-Flop JK
http://www.play-hookey.com/digital/jk_nand_flip-flop.html

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

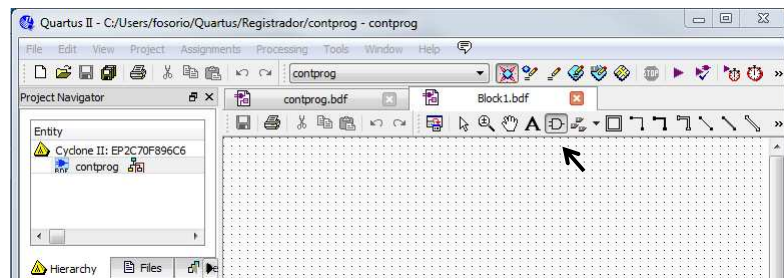
Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)

Quartus:

Blocos pré-definidos...

Primitives – Storage - jkff



Editor de Diagramas (BDF)

Selecionar: Symbol Tool 

7

Agosto 2008

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

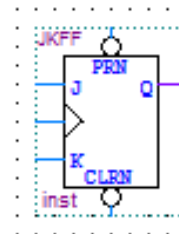
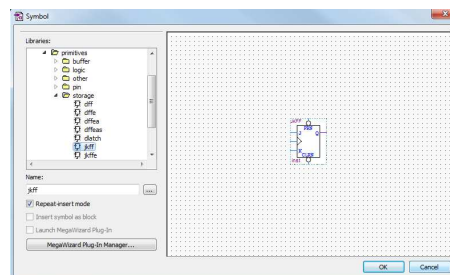
Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)

Quartus:

Blocos pré-definidos...

Primitives – Storage - jkff



Editor de Diagramas (BDF)

Selecionar: Symbol Tool 

8

Agosto 2008

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)

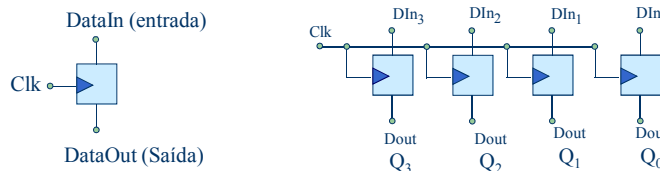
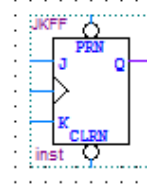
Um conjunto de FFs (Flip-Flops) pode formar um REGISTRADOR de 'n' Bits, onde podemos ter ainda sinais de:

CargaReg: Grava no registrador os dados vindos pelo barramento DataIn[N..0]

 Pode ser usado o CLK como sinal de controle de CargaReg

ResetaReg: Zera o conteúdo dos bits do Registrador (CLR/N)

LeReg: Habilita a leitura do Registrador (pode usar um MUX para esta função)



9

Agosto 2008

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Flip-Flop

Armazenando dados – Memórias de 1 bit

Memórias de 1 bit

Flip-Flop JK (J=Set / K=Reset)

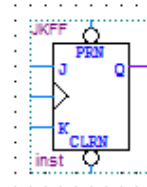
Um conjunto de FFs (Flip-Flops) pode formar um REGISTRADOR de 'n' Bits, onde podemos ter ainda sinais de:

CargaReg: Grava no registrador os dados vindos pelo barramento DataIn[N..0]

 Pode ser usado o CLK como sinal de controle de CargaReg

ResetaReg: Zera o conteúdo dos bits do Registrador (CLR/N)

LeReg: Habilita a leitura do Registrador (pode usar um MUX para esta função)



REGISTRADORES também são usados como:

Contadores: Sinal de Incremento (+1) para contagem

Registradores de Deslocamento: Shift Register para inserir ou retirar dados serialmente e para realizar operações de multiplicação e divisão inteira por 2 (shift left / shift right)

10

Agosto 2008

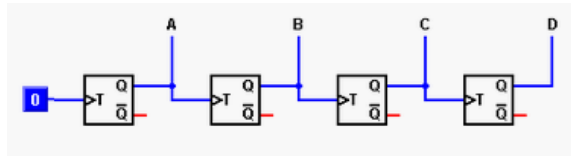
3. Flip-Flop: Contadores

Flip-Flop - Contador

Registrador: Contagem de 1 em 1

Principal aplicação:

- Registrador do “Contador de Programa” (PC: *Program Counter*) usado para guardar o endereço da próxima instrução a ser buscada na memória, decodificada e executada pelo processador
- Incrementar o valor de uma variável (mais simples que a soma realizada com o uso de uma ULA – Unidade Aritmética genérica)



Contador usando Flip-Flops do tipo T - Referência: (simulador)

<http://www.play-hookey.com/digital>

http://www.play-hookey.com/digital/ripple_counter.html

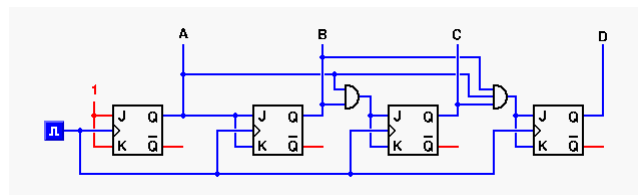
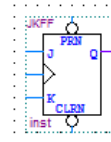
11

Agosto 2008

3. Flip-Flop: Contadores

Flip-Flop - Contador

Registrador: Contagem de 1 em 1



Contador usando Flip-Flops do tipo JK - Referência: (simulador)

<http://www.play-hookey.com/digital>

http://www.play-hookey.com/digital/synchronous_counter.html

Shift-register:

http://www.play-hookey.com/digital/shift-in_register.html

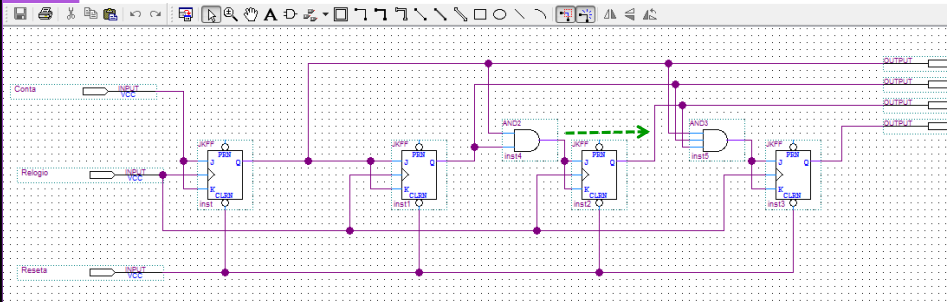
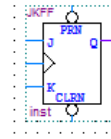
12

Agosto 2008

3. Flip-Flop: Contadores

Flip-Flop - Contador

Registrador: Contagem de 1 em 1
> Implementando um contador de 4 bits no Quartus

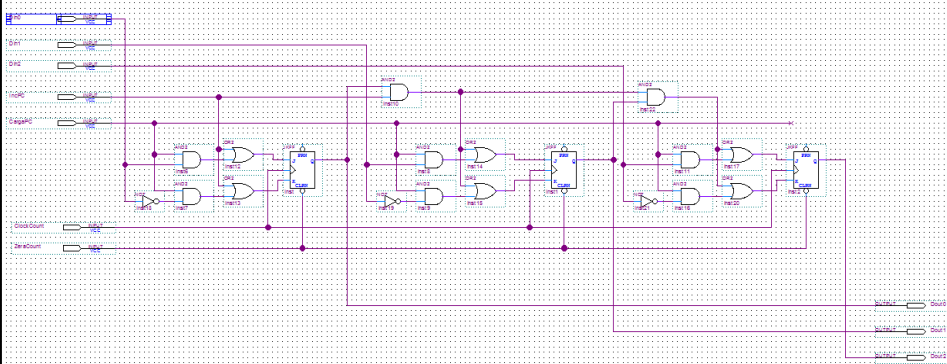
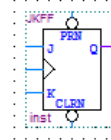


Porta AND final de 3 entradas pode ser substituída por uma porta AND de 2 entradas que recebe "em cascata" as saídas das portas AND anteriores (seta em verde)

3. Flip-Flop: Contadores

Flip-Flop - Contador

Registrador tipo PC: Contador de Programa
> Implementando um PC de 3 bits no Quartus
> Funções: Zerar (Reseta), Incrementa (+1), CargaPC (Setar Valor)



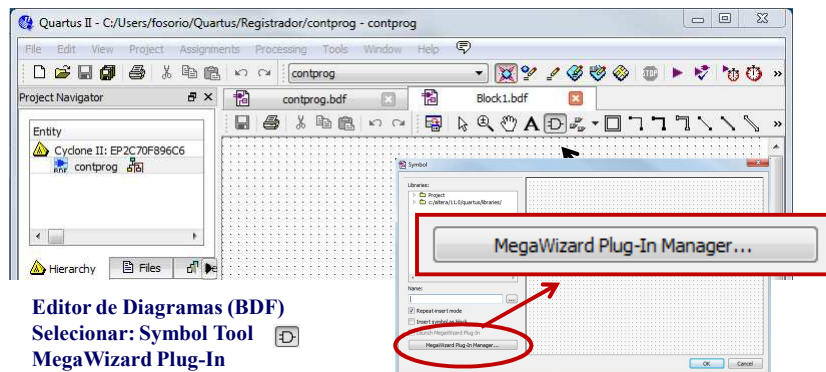
3. Flip-Flop: Memória de N Bits

Memória:

Armazenando dados – Memórias de N bits

Memórias RAM e ROM

Quartus:
 Blocos pré-definidos...
 MegaWizard Plug-In Manager



Editor de Diagramas (BDF)
 Selecionar: Symbol Tool
 MegaWizard Plug-In

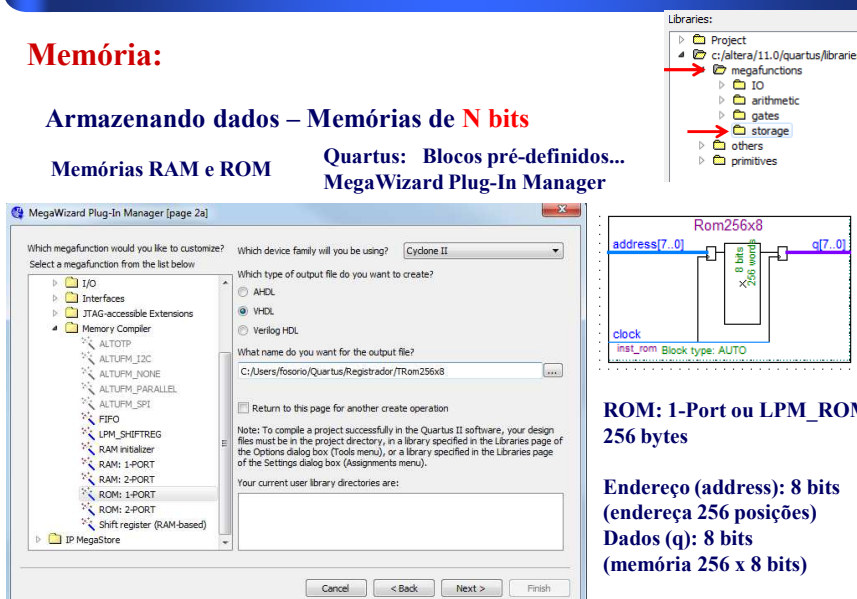
3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Memória:

Armazenando dados – Memórias de N bits

Memórias RAM e ROM

Quartus: Blocos pré-definidos...
 MegaWizard Plug-In Manager



ROM: 1-Port ou LPM_ROM
 256 bytes

Endereço (address): 8 bits
 (endereça 256 posições)
 Dados (q): 8 bits
 (memória 256 x 8 bits)

3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Memória:

Armazenando dados
Memórias ROM

N bits

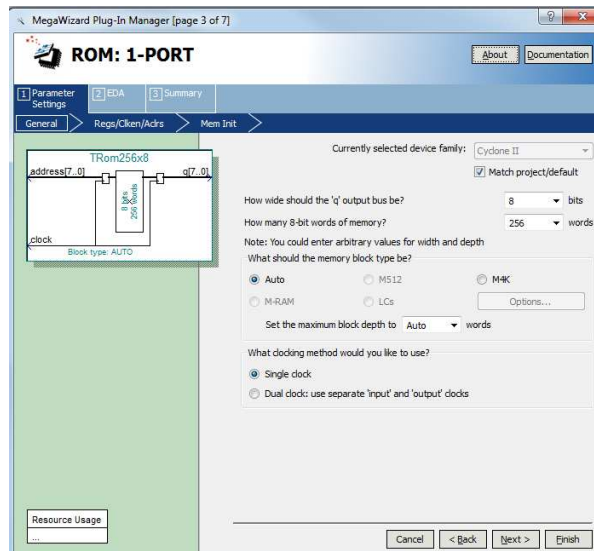
Exemplo:

256 posições de 8 bits

Quartus: Blocos pré-definidos
MegaWizard Plug-In Manager

19

Agosto 2008



3. Flip-Flop: Memória de 1 bit (RS, JK, Latch D)

Memória:

Armazenando dados
Memórias ROM

N bits

Exemplo:

256 posições de 8 bits

Carregando os dados
Gravados na ROM...
Arquivo .hex ou .mif

Descrição .hex em:

http://en.wikipedia.org/wiki/Intel_hex

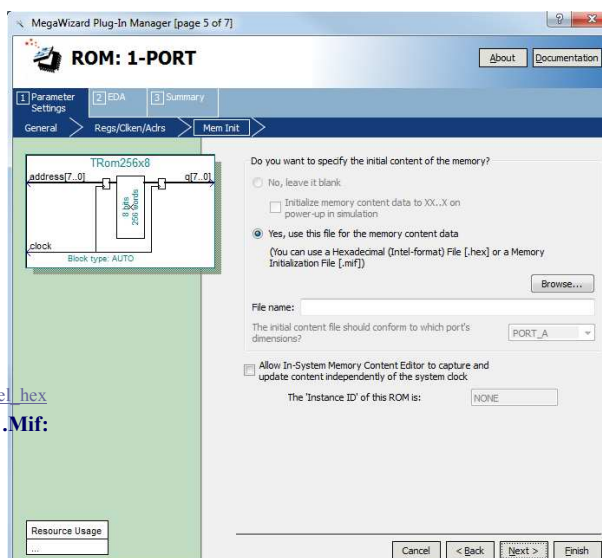
Gerando arquivos .Hex e .Mif:

Quartus => File – New -
Memory Files

Ver proposta de projeto final
Para mais detalhes sobre ROM

20

Agosto 2008





INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

LRM – Laboratório de Robótica Móvel

Web LRM: [Http://lrm.icmc.usp.br/](http://lrm.icmc.usp.br/)

Página pessoal: [Http://www.icmc.usp.br/~fosorio/](http://www.icmc.usp.br/~fosorio/)

E-mail: [fosorio \[at\] { icmc. usp. br , gmail. com }](mailto:fosorio@icmc.usp.br) – F.Osório

E-mail: [diogosoc \[at\] { icmc. usp. br }](mailto:diogosoc@icmc.usp.br) - Diogo Correa (PAE)

Disciplina de Laboratório de Elementos de Lógica Digital I [LELD1]

Web Disciplinas: [Http://www.icmc.usp.br/~fosorio/](http://www.icmc.usp.br/~fosorio/)

Web Wiki: [Http://wiki.icmc.usp.br/index.php/SSC-111](http://wiki.icmc.usp.br/index.php/SSC-111)

> Programa, Material de Aulas, Critérios de Avaliação,

> Material de Apoio, Trabalhos Práticos