Universidade de São Paulo - São Carlos,SP

Instituto de Ciências Matemáticas e de Computação

USP – ICMC – SSC

SSC0111 - Laboratório de Elementos de Lógica Digital

Professor responsável: *Fernando Santos Osório*
Estagiário PAE: Diogo Correa
Semestre: 2011/2
Horário: Quinta 16h20

E-mail: fosorio .at. icmc.usp.br
fosorio .at. gmail.com
Web: ***http://www.icmc.usp.br/~fosorio /***

TRABALHO PRÁTICO Nro. 01

Definição de 31/08/2011

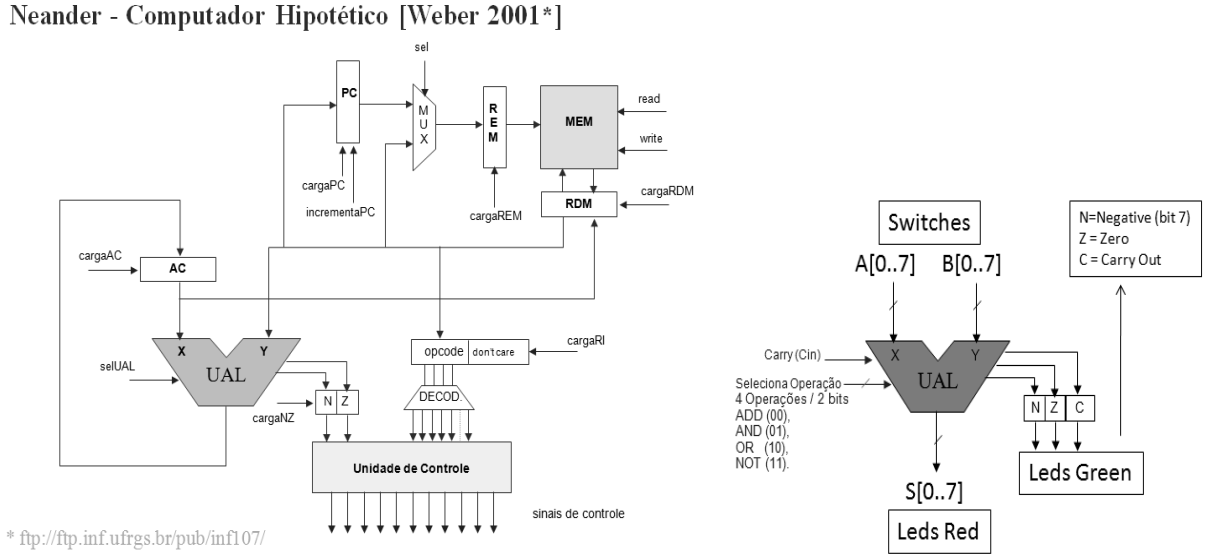
[Descrição Geral]

Os projetos serão desenvolvidos usando o software Quartus II da Altera, sendo implementados em FPGA através de *Block Diagrams* (BDF) e simulados usando *Vector Waveforms* (VWF). Além disto, o projeto deve ser implementado e testado na placa DE2-70 (FPGA Terasic-Altera) disponível no Laboratório. Os arquivos descrevendo o projeto serão entregues ao professor, conforme descrito mais abaixo (envio por e-mail de um .zip ou .rar contendo o diretório de projeto onde obrigatoriamente se encontram os arquivos .qpf, .qsf, .bdf, .pin, .vwf).

Este trabalho consiste em implementar, simular e executar dois projetos em FPGA:
Projeto 01 – Unidade Lógico Aritmética "simples" (ULA)
Projeto 02 – Circuito Combinacional

[Projeto 01 - ULA]

A parte 01 do trabalho consiste da implementação da ULA do computador didático Neander em FPGA. Para fins ilustrativos, segue abaixo o esquema completo do referido processador Neander:



A implementação da ULA do Neander em FPGA consiste de uma ULA de 8 bits (bits 0 a 7), onde podem ser executadas 4 operações básicas conforme a seleção: ADD (Soma=00), AND (E bit-a-bit=01), OR (Ou bit-a-bit=10) e NOT (Inverte bits=11).

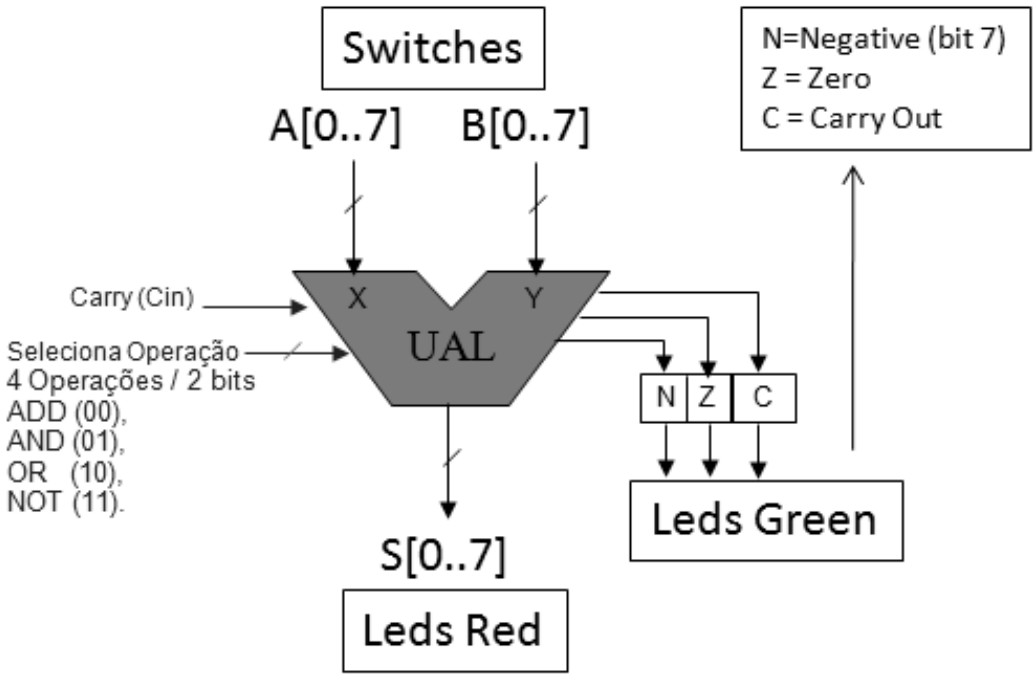
A operação ADD é selecionada com os 2 switches de "seleção da operação" na posição 00. A soma irá somar os 2 dados de entradas de 8 bits, A[0..7] e B[0..7], através do uso de somadores completos (*full-adder*) em cascata. Os somadores terão como entrada os operadores A e B, além do *Carry-In* (Cin). Como resultado da soma, será obtido um valor de 8 bits, S[0..7], que representa o resultado da operação. Além disto, os 3 leds irão indicar se este valor resultante é igual a zero (Z), se ele possui o bit 7 ligado (N), e se ocorreu um *Carry-Out* (C) no último bit, ou seja, um vai-um do bit 7. Note que a subtração poderá ser operada através da operação de soma, porém com números em representação de complemento de 2 (números negativos).

A operação de AND é selecionada com os 2 switches de "seleção da operação" na posição 01. Esta operação realiza um AND bit-a-bit entre os dados A[0..7] e B[0..7], resultando em S[0..7]. O resultado desta operação irá afetar os leds de Z (Zero) e N (Negative), mas no entanto não irá afetar o C (Carry-Out) pois não gera vai-um.

A operação de OR é selecionada com os 2 switches de "seleção da operação" na posição 10. Esta operação realiza um OR bit-a-bit entre os dados A[0..7] e B[0..7], resultando em S[0..7]. O resultado desta operação irá afetar os leds de Z (Zero) e N (Negative), mas no entanto não irá afetar o C (Carry-Out) pois não gera vai-um.

A operação de NOT é selecionada com os 2 switches de "seleção da operação" na posição 11. Esta operação realiza um NOT bit-a-bit (inversão de bits) dos dados de A[0..7], resultando em S[0..7]. O resultado desta operação irá afetar os leds de Z (Zero) e N (Negative), mas no entanto não irá afetar o C (Carry-Out) pois não gera vai-um. Note que as entradas B[0..7] não serão usadas nesta operação, pois o dado usado sempre será o dado contido em A[0..7].

Como entrada da ULA serão usados 16 switches indicando o valor de A[0..7] e B[0..7], e mais 2 outros indicando a seleção da operação da ULA (00, 01, 10 ou 11). Além deste, um *push-button* indicará o estado do *Carry-In* (Cin). Como saída da ULA serão usados 8 leds vermelhos para indicar o resultado da operação, e mais 3 leds verdes para representar o status N, Z e C. A figura abaixo apresenta o esquema da ULA.



Teste por simulação: o circuito da ULA implementado deverá ser simulado, onde o vetor de testes (vwf) deverá obrigatoriamente conter os dados conforme especificado a seguir. A ULA deverá ser testada para cada uma das ***4 operações usando como entrada os 2 dígitos finais dos Números USP dos alunos*** que compõem o grupo do trabalho. Por exemplo, para um aluno com o NUsp 1234567 e o outro com o NUsp 7654321, então devem ser testadas as operações: 67 ADD 21, 67 AND 21, 67 OR 21, NOT 67. Se um aluno fizer o trabalho sozinho, então use os 2 dígitos finais com os 2 dígitos iniciais de seu próprio NUsp. O arquivo VWF deve conter o vetor de teste com estes dados.

[Projeto 02 – Circuito Combinacional]

Para cada aluno da turma será especificada uma expressão booleana descrevendo um circuito combinacional de até 4 variáveis de entrada (A, B, C, D) e 2 variáveis de saída (S1 e S2, uma para cada aluno do grupo, com seu respectivo circuito), ou 1 única variável de saída (S1, no caso de um trabalho realizado de forma individual). Este circuito deve ser implementado no Quartus II, sendo simulado considerando as 4 possíveis combinações das entradas (16 vetores de entrada). Entregar o circuito e a simulação.

Os circuitos estão especificados na tabela abaixo, onde está indicado o Nro. USP de cada aluno, com a respectiva expressão booleana a ser implementada em FPGA.

1.	7696372	S = not ((A or B) and (C or D))
2.	7696271	S = (not A and not B) or (not C and not D)
3.	7656575	S = (not A or not B) and (not C or not D)
4.	7696393	S = ((A or B) and (C or D)) xor A
5.	7705005	S = ((A and B) or (C and D)) xor B
6.	7696409	S = not (A and B) or not (C and D)
7.	7656241	S = not (A or B) and not (C or D)
8.	7573580	S = (A xor B) and (C xor D)
9.	7573506	S = (A and B and C and D) or (A or B or C or D)
10.	7656168	S = (A or B or C or D) and (A and B and C and D)
11.	7696497	S = not ((not A) and (not B) and (not C) and (not D))
12.	7656620	S = not ((not A) or (not B) or (not C) or (not D))
13.	7656147	S = (A and B) xor (C and D)
14.	7696434	S = (A or B) xor (C and D)
15.	7151885	S = ((A xor B) xor A) and ((C xor D) xor C)
16.	7573486	S = not (A xor B) and (C or D)
17.	7696264	S = not (A xor B) or (C and D)
18.	7704971	S = not (A xor B xor C xor D)
19.	7573187	S = ((not A) and B) or ((not C) and D)
20.	6878226	S = (A or (not B)) and (C or (not D))
21.	7696330	S = (A and B and C and D) or (A xor B)
22.	7573472	S = (A or B or C or D) xor (A and B)
23.	7696559	S = (A xor D) and (B xor C)
24.	7656404	S = (A and B) xor (C or D)
25.	7696312	S = not (A and B) xor not (C and D)
26.	7704964	S = ((not A) and B) or (A and (not B)) or C or D

Ponto Extra:

* Será atribuído um ponto adicional (+0.5 na nota) para o grupo que além de apresentar os dados de saída da ULA com o uso dos LEDs, também usar o display de segmentos da placa DE2-70 para apresentar o valor final obtido na saída da operação.

==========

ENTREGA DO TRABALHO:

* Envie um e-mail com os arquivos do projeto completo de seu trabalho ao prof. Osório
  (incluir os arquivos que compõem seu projeto: .zip ou .rar + documentação)
  E-MAIL TO: **fosorio@gmail.com** (Enviar o original para este email)
  EMAIL CC: **work2usp@yahoo.com** (Enviar com cópia para este email)
  SUBJECT: **[SSC0111] TP01 <nomes_dos_alunos>** (Assunto do email)

* Escreva no corpo da mensagem de e-mail:
  NOME: <nome completo_aluno_1> + <Nro. USP_Aluno1>
  NOME: <nome completo_aluno_2> + <Nro. USP_Aluno2>
  INFORMAÇÕES SOBRE O PROJETO: <projeto desenvolvido>,
  <informações que julgar necessárias para a avaliação e teste do seu projeto>

ANEXOS – Atenção: A mensagem deve conter como anexo arquivos .zip ou .rar

INCLUIR: O anexo (zip ou rar) deve conter o diretório do projeto do Quartus II (procure colocar todos arquivos do projeto neste mesmo diretório). Este arquivo deve obrigatoriamente conter os seguintes arquivos: .qpf e .qsf (projeto quartus), .bdf (block diagram file – esquemático do projeto), .pin (atribuição de pinagem), e dois arquivos .vwf (vector waveform file) contendo o vwf de entrada da simulação e o vwf do report (.sim.vwf) gerado como saída da simulação (se for possível adicionar também uma captura de tela da saída da simulação).

NÃO INCLUIR: Não devem ser compactados e incluídos em hipótese alguma junto ao zip/rar, arquivos como .exe, .com, .bat, pois o sistema de e-mail não aceita o envio de arquivos com este conteúdo.

Atenção: O professor irá confirmar o recebimento dos arquivos. Se não receber uma confirmação é porque seu trabalho não chegou! Neste caso, contate o professor IMEDIATAMENTE, pois caso contrário seu trabalho será considerado como "não entregue".

* Entregar até a data indicada no Site da Disciplina / Wiki ICMC / SSC0111(fosorio)
  http://www.icmc.usp.br/~fosorio/ (Trabalhos Práticos)

===== THAT'S ALL FOLKS !!! =====