Departamento de Engenharia Elétrica, PUC-Rio
ELE 1448 – Computação Digital
Prof. Marco Aurélio C. Pacheco

G3-Projeto

Projeto: Simulação e Aplicação de um Computador com as seguintes características:

- Palavra de 4 bits
- Espaço de endereçamento direto de 16 posições de memória
- Instruções ocupam 1 ou 2 palavras consecutivas de 4 bits na memória. Formato de instrução:

Opcode (4 bits)
Operando (4 bits) quando houver

- as instruções a serem implementadas na prova são apenas as mostradas abaixo:

Inst.	Opcode	Operand	Comentário
bneq	0000	*addr*	Salto condicional para adrr se Z=0 (branch not equal)
add AC, (A0) +	0001		soma acumulador com o conteúdo da memória apontada pelo registro A0; incrementa A0
out	0010	*#porta*	Transfere conteúdo do acumulador para a porta de saída indicada pelo número #porta

A arquitetura do data path da CPU é fornecida.

1. Todas as entidades abaixo são de 4 bits:
   - Acumulador (AC) mais um bit de vai um CARRY
   - Registro de endereçamento (contador) (A0)
   - Programa Counter (contador) (PC)
   - Instruction Register (IR)
   - Porta de Saída ligada a um Display ASCII.
   - ALU 4 bits usando a função soma

2. O sinal de RESET deve fazer PC=10H, AC=0 e A0=0H (corresponde ao endereço da Tabela no programa).
3. Barramentos: 2 barramentos de 4 bits (D0-D3) e (B0-B3)
4. Acesso aos barramentos: todos os registros podem ser acessados para leitura e/ou escrita pelos 2 barramentos; implemente apenas os acessos que forem necessários.
5. ALU: É preciso usar ALU nesse projeto.
6. Memória: use ROM (16x4, despreze os 4 bits mais significativos da ROM 32x8): Ligação aos barramentos: Dados (I/O) = (D0-D3); Endereço = (B0-B3).

a) Faça na folha de solução um diagrama do data path da CPU conforme já especificado acima, incluindo: registros, memória, barramento, ligação com a unidade de controle, ligações com E/S, os 3-states e os sinais de enable e clocks de registros necessários ao seu projeto. (1,5)

b) Usando a linguagem RTL, detalhe na folha quadriculada as operações de controle para a execução do fetch e das instruções mostradas na folha de solução, indicando os estados Ti e os sinais acionados. (4 itens x 0,5=2,0)

c) Implemente o Data Path (memória, registros, barramento, porta de saída ligada a Display ASCII e circuitos de acesso). Crie um painel de controle com os seguintes itens (todos juntos no canto esquerdo da tela): um LED 7-seg ligado a cada registro, uma chave de RESET e um display ASCII com o pino5=pino6='0' e pino7= '1'. (1,5)

d) Projete e programe a Unidade de Controle para esta CPU (fixa ou microprogramada). (3,0)

e) Usando o relógio central conectado à CPU, faça o seu computador executar, continuamente, o programa abaixo. Tanto o programa quanto os dados da tabela devem ser armazenados por você na sua ROM de programa (2,0)

```
        ORG   10H
10H     loop: add AC, (A0) +          ; lê próximo dado da tabela
11H           bneq loop               ; permanece em loop
13H           out FH                  ; manda para display cuja porta tem número 15 decimal
15H           bneq loop               ; permanece em loop
        ORG   0H
Tabela: DS    16 ; 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1   ; dados hexadecimais armazenados na ROM
```

- ***Entrega do Projeto: Impresso e em mídia com 2 arquivos: um arquivo com o Projeto.ckt e um arquivo Word ou EXCEL com o Detalhamento do Conjunto de Instruções (Tabela de Controle), Diagrama em Blocos da CPU e o seu Programa Aplicaçãoem Assembly comentado.***
- ***Apresentação: apresentar ao monitor no laboratório o projeto funcionando.***

diagrama do data path

Ti	RTL	sinais																				Fetch/comentários