

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

**Engenharia Informática e de Computadores
e
Engenharia de Electrónica e Telecomunicações e de Computadores**



**1º Trabalho Prático
de
Arquitectura de Computadores**

*Projecto de uma
Unidade Central de Processamento*

19 de Março de 2007

Objectivos

O trabalho tem por objectivo principal desenvolver a arquitectura de uma unidade central de processamento (CPU) para execução de um conjunto de instruções *assembly*. Para tal, o aluno irá utilizar as técnicas e os modelos apresentados nas aulas teóricas para o projecto deste tipo de arquitecturas, nomeadamente as técnicas de codificação de instruções, o modelo *basic schemata* e a síntese da arquitectura a partir da representação *basic schemata*.

Descrição Funcional

O CPU a projectar deve seguir uma arquitectura típica formada por uma unidade de processamento e uma unidade de controlo. No projecto da arquitectura, deve ter em conta os aspectos seguintes:

- Bus de dados interno e de acesso à memória de 8 bits;
- Capacidade de endereçamento de memória de código de 2K posições;
- Capacidade de endereçamento de memória de dados de 256 posições;
- Um banco de 4 registos;
- Uma pilha localizada em memória de dados. O topo da pilha é identificado pelo registo SP (registo interno do CPU). O registo SP é iniciado com o valor 10h após inicialização do CPU (*reset*);
- Uma ALU e um registo A (acumulador). A ALU é implementada como um circuito combinatório que, para além do resultado das operações, gera uma *flag* CY (*carry/borrow*) que reflecte o arrasto da última operação aritmética realizada;
- Um conjunto de instruções que deve ser um de três possíveis (ver anexo A);

Desenvolvimento

No desenvolvimento do projecto, considere os passos seguintes:

- a) Escolha do conjunto de instruções;
- b) Codificação das instruções;
- c) Representação da estrutura do CPU em *basic schemata* (EFI e ESA);
- d) Desenho do diagrama de blocos da unidade de processamento (módulo funcional) e da máquina de estados da unidade de controlo (módulo de controlo) do CPU a partir da representação em *basic schemata*;
- e) Desenvolvimento de um programa de simulação da estrutura proposta com base na representação *basic schemata*. Neste ponto, deve utilizar um programa tipo de simulação, cuja estrutura é dada ([programa](#)), que terá de modificar por forma a suportar a simulação do conjunto de instruções do CPU;
- f) Execução das funções ACUMULA() e SOMA() no CPU desenvolvido. Deve começar por traduzir as funções para linguagem *assembly*, com base no conjunto de instruções do seu

CPU. Deve depois converter a descrição para código máquina, carregá-la na memória do simulador e executá-la por simulação.

```
char ACUMULA(char arrayA[]){  
    c = 0;  
    for i in 0 to 99  
        if (arrayA[i] ≠ i)  
            c = SOMA(A[i], c)  
}  
  
char SOMA(char elemento, char c){  
    return (elemento + c)  
}
```

Nota: os parâmetros são passados por registo ou pela pilha

Avaliação

O trabalho deve ser realizado em grupo, conta para a avaliação da disciplina e está sujeito a discussão final. A apresentação do trabalho decorre em laboratório em data a combinar com o respectivo docente. No final do projecto, cada grupo deverá entregar um relatório do trabalho desenvolvido, no qual conste:

- Os resultados obtidos durante todas as fases de desenvolvimento;
- Listagens dos programas de teste;
- Conclusões.

Anexo

Conjunto de Instruções 1

STA	addr	$(addr) \leftarrow A;$
LDA	addr	$A \leftarrow (addr);$
MOV	Rn,A	$Rn \leftarrow A;$
MOV	A,#const8	$A \leftarrow const8;$
MOV	A,Rn	$A \leftarrow Rn;$
MOV	A,@Rn	$A \leftarrow (Rn);$
ADD	A,Rn	$A \leftarrow A + Rn;$
SUB	A,Rn	$A \leftarrow A - Rn;$
JNZ	A,offset8	if $(A \neq 0)$ $PC \leftarrow PC + offset8;$
DJNZ	A,offset8	$A = A - 1;$ if $(A \neq 0)$ $PC \leftarrow PC + offset8;$
PUSH	A	$SP = SP + 1; (SP) \leftarrow A;$
POP	A	$A \leftarrow (SP); SP = SP - 1;$
SJMP	offset8	$PC \leftarrow PC + offset8;$
LCALL	c_addr	$SP = SP + 1; (SP) \leftarrow PC; PC \leftarrow c_addr;$
RET		$PC \leftarrow (SP); SP = SP - 1;$

Conjunto de Instruções 2

MOV	addr,Rn	$(addr) \leftarrow Rn;$
MOV	Rn,addr	$Rn \leftarrow (addr);$
MOV	Rn,#const8	$Rn \leftarrow const8;$
MOV	Rn,@Rm	$Rn \leftarrow (Rm);$
ADD	Rn,Rm	$Rn \leftarrow Rn + Rm;$
SUB	Rn,Rm	$Rn \leftarrow Rn - Rm;$
JNZ	Rn,offset8	if $(Rn \neq 0)$ $PC \leftarrow PC + offset8;$
DJNZ	Rn,offset8	$Rn = Rn - 1;$ if $(Rn \neq 0)$ $PC \leftarrow PC + offset8;$
PUSH	Rn	$(SP) \leftarrow Rn; SP = SP + 1;$
POP	Rn	$SP = SP - 1; Rn \leftarrow (SP);$
SJMP	offset8	$PC \leftarrow PC + offset8;$
LCALL	c_addr	$(SP) \leftarrow PC; PC \leftarrow c_addr; SP = SP + 1;$
RET		$SP = SP - 1; PC \leftarrow (SP);$

Conjunto de Instruções 3

MOV	addr, Rn	$(addr) \leftarrow Rn;$
MOV	Rn, addr	$Rn \leftarrow (addr);$
MOV	Rn,@Rm	$Rn \leftarrow (Rm);$
MOV	Rn,Rm	$Rn \leftarrow Rm;$
JNZ	Rn,offset8	if $(Rn \neq 0)$ $PC \leftarrow PC + offset8;$
DJNZ	Rn,offset8	$Rn = Rn - 1;$ if $(Rn \neq 0)$ $PC \leftarrow PC + offset8;$
PUSH	Rn	$SP = SP + 1; (SP) \leftarrow Rn;$
POP	Rn	$Rn \leftarrow (SP); SP = SP - 1;$
SJMP	offset8	$PC \leftarrow PC + offset8;$
LCALL	c_addr	$SP = SP + 1; (SP) \leftarrow PC; PC \leftarrow c_addr;$
RET		$PC \leftarrow (SP); SP = SP - 1;$
ADD	Rn,Rm,Rk	$Rn \leftarrow Rm + Rk;$
SUB	Rn,Rm,Rk	$Rn \leftarrow Rm - Rk;$

const8 – constante de 8 bits; **offset8** – deslocamento relativo de 8 bits ([-128, +127])