

ELE1717 - sistemas digitais - Semana 3

(o líder é o primeiro aluno de cada grupo)

Grupo 01

Líder	Matricula	Nome
	20190154363	GABRIEL DA SILVA LIMA
	20200150248	PAULO EDUARDO SILVA DE OLIVEIRA
	20200150257	SAMARA REVOREDO DA SILVA
•	20200150266	SAMUEL VICTOR MACIEL DA SILVA
	20200001121	WESLEY WAGNER VARELA SOUZA

Grupo 02

Líder	Matricula	Nome
	20190154022	ARTHUR FELIPE RODRIGUES COSTA
	20200001023	ERIKA COSTA ALVES
	20200001079	LUCAS GUALBERTO SANTOS RIBEIRO
	20200150210	LUIZ VITOR CLEMENTINO
•	20180010172	RAFAEL DE MEDEIROS MARIZ CAPUANO

Grupo 03

Líder	Matricula	Nome
	20190002620	ALISSON GABRIEL LUCAS DA SILVA
	20200001050	KAIKE CASTRO CARVALHO
•	20200001088	MATEUS DE ASSIS SILVA
	2015093037	MAURICIO ALVES DA SILVA
	20190001883	WILLIAN MOURA GONDIM DE FREITAS

Grupo 04

Líder	Matricula	Nome
	20180153809	CAIO LINS MACHADO
	2016017479	HILO DE OLIVEIRA GOES
	20200150195	LUCAS AUGUSTO MACIEL DA SILVA
•	20200150201	LUCAS BARBOSA DE MEDEIROS
	20170040454	LUCAS DA SILVA LEONCIO

Grupo 05

Líder	Matricula	Nome
	20200001014	ELKE SAMANTHA DA SILVA DOMINGOS
	20200001060	LEANDRO DE SOUZA RODRIGUES
	20200150239	MOHAMAD SADEQUE ABOU ALI
	20190001909	PEDRO HENRIQUE DE OLIVEIRA FREIRE
•	20200150275	VINICIUS NASCIMENTO DE AZEVEDO

Disciplina: ELE1717 - Sistemas Digitais
Aluno:

Período: 2020.2
Problema: 02

1- Projete um circuito integrado para um sistema digital que implemente uma unidade central de processamento (CPU) de uso geral com 16 instruções. A CPU deverá possuir um projeto RTL cuja a estrutura básica está de acordo com a Figura 1 e com a descrição do seu conjunto de instruções apresentada na Tabela 1.

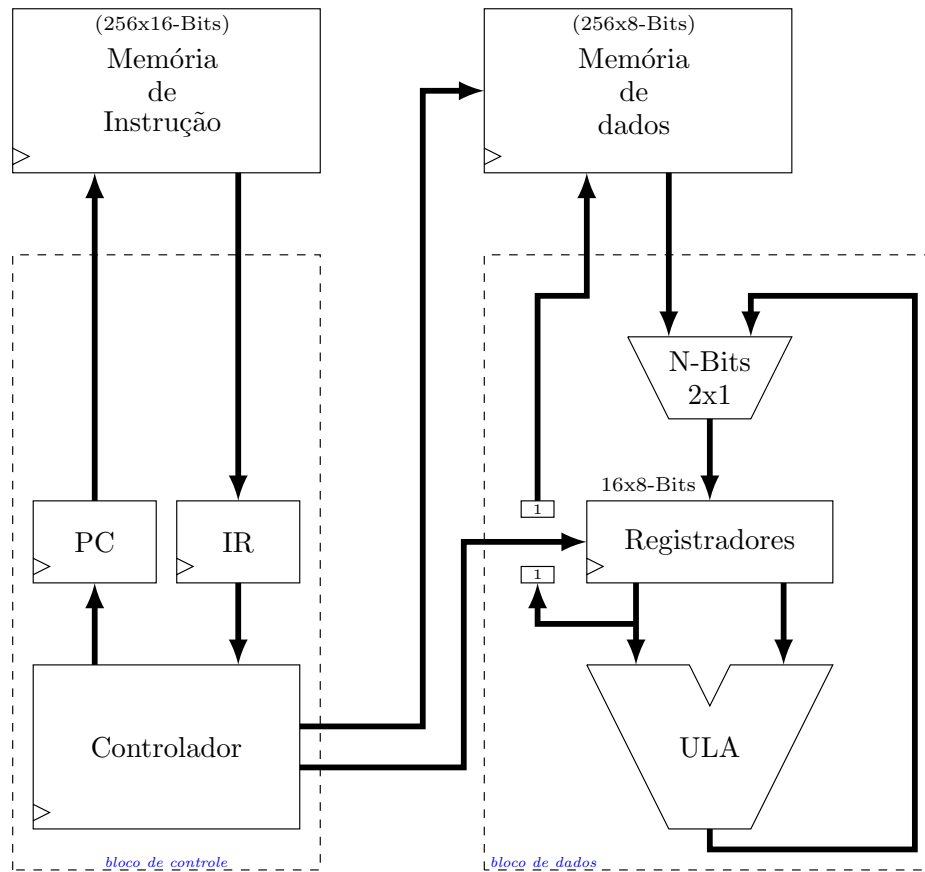


Figura 1: Estrutura básica do projeto RTL de uma CPU de uso geral

Oper.	Classe	Opcod	4bits	4bits	4bits	Descrição	Carry	ULA
HLT	Controle	0000	-	-	-	$PC_{k+1} = PC_k$		
LDR	Dados	0001	A	addr[7..4]	addr[3..0]	$Reg[A] \leftarrow Mem_D[addr]$		
STR	Dados	0010	A	addr[7..4]	addr[3..0]	$Mem_D[addr] \leftarrow Reg[A]$		
MOV	Dados	0011	-	B	C	$Reg[B] \leftarrow Reg[C]$		
ADD	ULA	0100	A	B	C	$Reg[A] \leftarrow Reg[B] + Reg[C]$	•	•
SUB	ULA	0101	A	B	C	$Reg[A] \leftarrow Reg[B] - Reg[C]$	•	•
AND	ULA	0110	A	B	C	$Reg[A] \leftarrow Reg[B] \text{ AND } Reg[C]$	•	•
OR	ULA	0111	A	B	C	$Reg[A] \leftarrow Reg[B] \text{ OR } Reg[C]$	•	•
NOT	ULA	1000	A	-	C	$Reg[A] \leftarrow \text{NOT } Reg[C]$	•	•
XOR	ULA	1001	A	B	C	$Reg[A] \leftarrow Reg[B] \text{ XOR } Reg[C]$	•	•
CMP	ULA	1010	A	B	C	$Reg[A] \leftarrow \text{CMP}(Reg[B], Reg[C])$	•	•
JMP	Salto	1011	-	value[7..4]	value[3..0]	$PC_{k+1} = \text{value}$		
JNC	Salto	1100	-	value[7..4]	value[3..0]	$PC_{k+1} = \text{value}, \text{ if carry}=0$		
JC	Salto	1101	-	value[7..4]	value[3..0]	$PC_{k+1} = \text{value}, \text{ if carry}=1$		
JNZ	Salto	1110	-	value[7..4]	value[3..0]	$PC_{k+1} = \text{value}, \text{ if ULA} \neq 0$		
JZ	Salto	1111	-	value[7..4]	value[3..0]	$PC_{k+1} = \text{value}, \text{ if ULA}=0$		

Tabela 1: Conjunto de instruções da CPU de uso geral.

Funcionamento do sistema:

O circuito integrado deverá ser capaz, de forma autônoma, executar um conjunto de rotinas pré-definidas na memória de instrução. Cada elemento da memória de instrução pode receber um dado de 16 bits referente a uma das 16 instruções (ver Tabela 1) disponíveis. As rotinas pré-definidas pelo usuário que serão carregadas na memória de instrução pode movimentar ou manipular dados de 8 bits. A manipulação sempre corre sobre dados salvos em um banco de registradores, já a movimentação ocorre sobre dados salvos no banco de registradores ou na memória de dados. Os dados iniciais utilizados pelas rotinas pré-definidas podem ser salvos pelo usuário na memória de dados. O registrador PC aponta para o endereço da memória de instrução que deverá ser lido, enquanto o registrador IR recebe o conteúdo do dado, proveniente da memória de instrução, que se encontra no endereço apontado pelo registrador PC. O bloco controlador é quem controla o endereçamento da memória de dados e do banco de registradores. O multiplexador (MUX) e a unidade lógica aritmética (ULA) são blocos puramente combinacionais. Por fim, o banco de registradores tem sua entrada conectada ao MUX e suas saídas conectadas a ULA e a memória de dados.

Para comprovar o correto funcionamento do circuito integrado deve-se elaborar dois códigos fontes para serem executados (individualmente):

- 1 - Desenvolva um código fonte para implementar um conversor que transforme um número binário de 8 bits em um número decimal (padrão ASCII para cada dígito).
- 2 - Desenvolva um código fonte para multiplicar dois números inteiros de 8 bits (resultado 16 bits).

É importante no projeto:

- Na semana de projeto é importante estudar projeto RTL, máquinas de estado (MDE) de alto nível, memórias, arquitetura de computadores e Assembly;
- O projeto será realizado através da especificação de uma MDE de alto nível, do projeto RTL estruturado (blocos de controle e de dados);
- Na semana de projeto não é necessário elaborar qualquer código fonte;
- O relatório do projeto deverá conter todas as especificações realizadas de tal forma que permita o leitor implementar o projeto;

É importante na implementação:

- Na semana de implementação é importante estudar MDE e memórias em VHDL, além disso, implementação de laços em Assembly;
- A implementação consiste no desenvolvimento de todos os códigos fonte necessários;
- O relatório da implementação deverá conter os diagramas do projeto corrigidos, se necessário for, e as indicações de correções realizadas no projeto;
- Para comprovar o funcionamento podem ser elaboradas simulações, as quais devem estar detalhadas no relatório e em vídeo;

Referências:

1. Livros de VHDL e sistemas digitais;
2. Livros de arquitetura de computadores;