



Laboratório de Circuitos Lógicos (CIC0231)

- Projeto Final -

ZEPTOPROCESSADOR-III DE 16 BITS

1. INTRODUÇÃO:

Os processadores são os circuitos digitais mais complexos que temos hoje em dia, principalmente os de alto desempenho. O conceito por trás de um processador envolve a leitura de uma instrução de uma memória, decodificar a instrução, executá-la e voltar a ler a próxima instrução. O presente projeto desafia o aluno a desenvolver um processador programável simples, o ZeptoProcessador-III, com apenas 13 instruções.

2. OBJETIVOS

Apresentar ao aluno os conceitos básicos que envolvem o projeto de processadores usando circuitos lógicos. O aluno deverá implementar um ZeptoProcessador-III de 16 bits capaz de executar programas com as seguintes instruções:

- 1) addi: Adição com imediato
- 2) subi: Subtração com Imediato
- 3) andi: AND bitwise com Imediato
- 4) ori: OR bitwise com Imediato
- 5) xori: XOR bitwise com Imediato
- 6) beq: Salto Condicional se igual
- 7) bne: Salto Condicional se diferente
- 8) ble: Salto Condicional menor ou igual (Signed)
- 9) bleu: Salto Condicional menor ou igual (Unsigned)
- 10) bgt: Salto Condicional maior que (Signed)
- 11) bgtu: Salto Condicional maior que (Unsigned)
- 12) jal: Salto incondicional ao endereço
- 13) jalr: Salto incondicional ao registrador

E executar programas com até 65536 instruções.

3. METODOLOGIA

Todo o sistema digital deve ser implementado como um projeto no software de simulação Deeds, com a interação com o usuário feita através do uso de uma memória ROM para armazenar o programa, botão de reset e sinal de clock, que pode ser manual ou automático.

O projeto deve ser baseado em interligação configurável de blocos funcionais de acordo com a instrução a ser executada.

O ZeptoProcessador-III gerado deverá ser testado pela execução de uma bateria de programas de testes.



4. ESPECIFICAÇÕES DO SISTEMA DIGITAL

O ZeptoProcessador-III deve possuir:

- i) Uma memória de instruções capaz de armazenar até 65536 instruções de 32 bits cada uma.
- ii) Um registrador PC de 16 bits que indica o endereço na memória da instrução executada.
- iii) Um banco de 16 registradores R0...R15 que podem armazenar números de 16 bits.
O registrador R0 possui o valor fixo 0.
- iv) Uma Unidade Lógico-Aritmética de 16 bits.

As instruções possuem 32 bits de tamanho codificados com os seguintes campos:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rb				Ra				Rd				Opcode			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Imediato															

Ra: 4 bits que definem o registrador Ra (de R0 a R15)

Rb: 4 bits que definem o registrador Rb (de R0 a R15)

Rd: 4 bits que definem o registrador Rd (de R0 a R15)

Imediato: Número de 16 bits

OpCode: são 4 bits que definem a instrução a ser executada

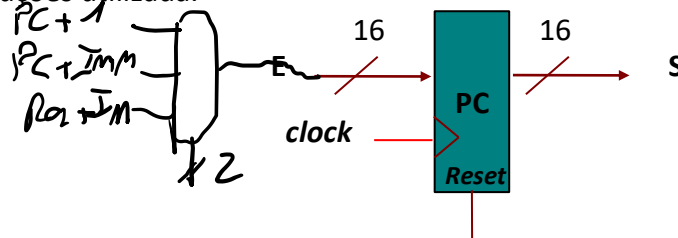
OpCode	Mnemônico	Nome	Operação
0000	addi Rd, Ra, Rb, Imm	Soma com imediato	$Rd = Ra + Rb + Imm$
0001	subi Rd, Ra, Rb, Imm	Subtração com imediato	$Rd = Ra - Rb - Imm$
0010	andi Rd, Ra, Rb, Imm	And bitwise com imediato	$Rd = Ra \& Rb \& Imm$
0011	ori Rd, Ra, Rb, Imm	Or bitwise com imediato	$Rd = Ra Rb Imm$
0100	xori Rd, Ra, Rb, Imm	Xor bitwise com imediato	$Rd = Ra \oplus Rb \oplus Imm$
0101	beq Ra, Rb, Imm	Salto se igual	$Ra == Rb ? PC = PC + Imm : PC = PC + 1$
0110	bne Ra, Rb, Imm	Salto se diferente	$Ra != Rb ? PC = PC + Imm : PC = PC + 1$
0111	ble Ra,Rb,Imm	Salto se menor ou igual	$Ra \leq Rb ? PC = PC + Imm : PC = PC + 1$ Ra e Rb considerados com sinal
1000	bleu Ra, Rb, Imm	Salto se menor ou igual unsigned	$Ra \leq Rb ? PC = PC + Imm : PC = PC + 1$ Ra e Rb considerados sem sinal
1001	bgt Ra, Rb, Imm	Salto se maior	$Ra > Rb ? PC = PC + Imm : PC = PC + 1$ Ra e Rb considerados com sinal
1010	bgtu Ra, Rb, Imm	Salto se maior unsigned	$Ra > Rb ? PC = PC + Imm : PC = PC + 1$ Ra e Rb considerados sem sinal
1011	jal Rd,Imm	Salto incondicional ao endereço	$Rd = PC + 1$ e $PC = PC + Imm$
1100	jalr Rd,Ra,Imm	Salto incondicional ao registrador	$Rd = PC + 1$ e $PC = Ra + Imm$



Geralmente, os processadores são compostos dos seguintes elementos básicos que devem ser projetados pelo aluno:

1) Registrador PC

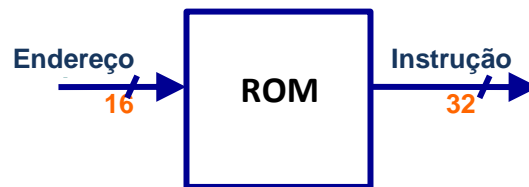
Trata-se de um registrador (use o modelo do Deeds) que armazena o endereço da instrução a ser executada. O número de bits deste registrador depende da quantidade de memória de instruções utilizada.



O dado de entrada E é escrito no registrador PC quando ocorrer a subida da borda de clock. Caso Reset=1, ao vir a borda de subida de clock o registrador é resetado, isto é, PC=0x0000.

2) Memória de Instruções

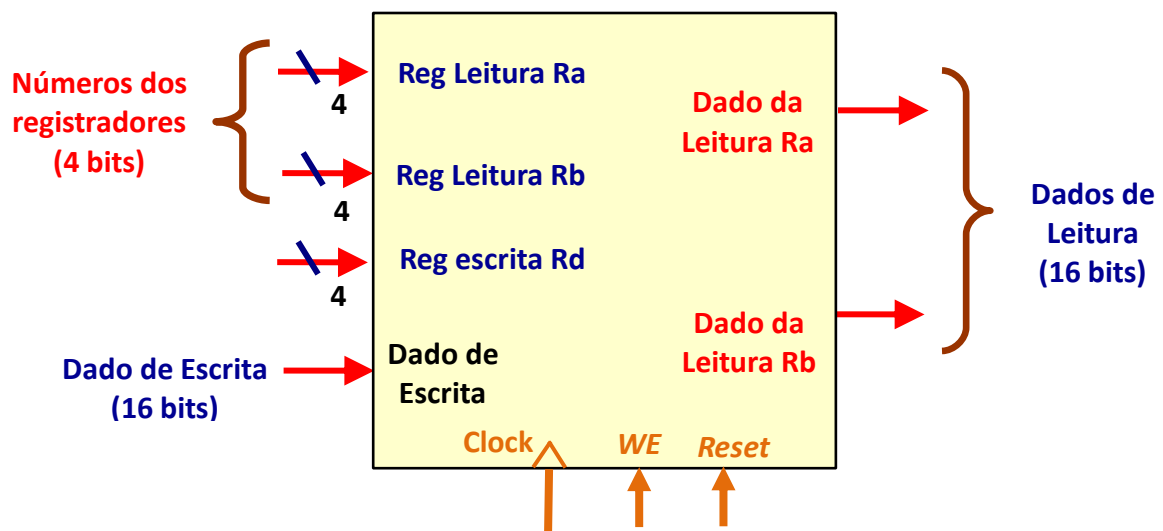
Consiste de uma memória do tipo ROM (use o modelo do Deeds) que deve ser programada com o código binário do programa do usuário. Dica: Use 2 módulos de ROM de 16 bits.



3) Banco de Registradores

Para a execução das instruções tem-se disponível um banco de 16 registradores de 16 bits cada um. O Banco de Registradores deve possibilitar a leitura de 2 registradores Ra e Rb simultaneamente e a escrita no registrador Rd se WE=1 e vier a borda de subida do clock.

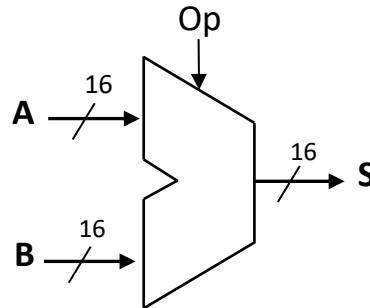
Caso Reset=1 na borda de subida do clock, todos os 16 registradores são resetados.





4) Unidade Lógico-Aritmética

Circuito digital (pode usar a ULA que tem pronta no Deeds) capaz de realizar as operações de soma e subtração de números de 16 bits em complemento de 2. A operação deve ser selecionável por um sinal Op de 3 bits (000-add, 001-sub, 010-and, 011-or, 100-xor).

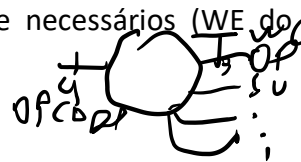


5) Comparador ==, !=, <= e > com sinal e sem sinal

Projete um circuito combinacional que receba 2 números A e B de 16 bits e indique em dois sinais de saída se $A \leq B$ e se $A == B$. Porém é necessário identificar se os números devem ser considerados com sinal ou sem sinal, assim acrescente um sinal de controle SU, onde $SU=0$ indica comparação sem sinal e $SU=1$ indica comparação com sinal.

6) Bloco de Controle

Projete um circuito combinacional que recebe na entrada o OpCode da instrução a ser executada e gere todos os sinais de controle necessários (WE do BR, Op da ULA, SU do Comparador e seleção de multiplexadores).



7) Sinais de Monitoramento

Acrescente ao seu processador displays de 7 segmentos hexadecimais que monitore os sinais: PC, Instrução, Ra e Rb.

5. EXEMPLOS DE PROGRAMAS

Em todos os programas exemplos aqui apresentados é pressuposto que o ZeptoProcessador-II inicia resetado (PC=0x0000 e Registradores=0x0000)

Exemplo 1: R1 = Contador de -16 a 16

Endereço	Código hexadecimal	Instrução	Comentário
0x0000	0x0010 0011	subi R1,R0,R0,16	R1=-16
0x0001	0x0010 0020	addi R2,R0,R0,16	R2=16
0x0002	0x0001 0221	subi R2,R2,R0,1	R2=R2-1 devido ao =
0x0003 Loop:	0x0001 0110	addi R1,R1,R0,1	R1=R1+1
0x0004	0xFFFF 2107	ble R1,R2,-1	R0<=R1? Loop : Next
0x0005 Fim:	0x0000 1105	beq R1,R1,0	J Fim - mostra R1



Exemplo 2: R3 = Soma dos números ímpares de 0 a 15

Endereço	Código hexadecimal	Instrução	Comentário
0x0000	0x0001 0010	addi R1,R0,R0,1	R1=1 primeiro ímpar
0x0001	0x000F 0020	addi R2,R0,R0,15	R2=15 valor final
0x0002	0x0001 0220	addi R2,R2,R0,1	R2=R2+1 (devido ao =)
0x0003 Loop:	0x0004 2108	bleu R2,R1,4	R2<=R1 ? Fim : Next
0x0004	0x0000 1330	addi R3,R3,R1,0	R3=R3+R1 somatório
0x0005	0x0002 0110	addi R1,R1,R0,2	R1=R1+2 próximo ímpar
0x0006	0xFFFF 0005	beq R0,R0,-3	J Loop
0x0007 Fim:	0x0000 3305	beq R3,R3,0	J Fim - mostra R3

6. PROGRAMA A DESENVOLVER

Elabore os algoritmos e implemente no ZeptoProcessador-III, onde os programas iniciam com:

Endereço	Código hexadecimal	Instrução	Comentário
0x0000	0x0000 0010	addi R1,R0,R0,0	R1 = 0 Resultado
0x0001	0xFFFF 0020	addi R2,R0,R0,X	R2 = X
0x0002	0/YYYY 0030	addi R3,R0,R0,Y	R3 = Y
0x0003	0x0002 00FB	jal R15, Proc	R15=0x0004 PC=Proc
0x0004 Fim:	0x0000 1105	beq R1,R1,0	J Fim - mostra R1
0x0005 Proc:
....
0xZZZZ	0x0000 0F0C	jalr R0,R15,0	Retorna resultado em R1

- 1) (0,5) Multu.drs: $R1 = R2 \times R3$: Multiplicação de dois números sem sinal ($R2 < 256$ e $R3 < 256$)
 - a. Filme o funcionamento
 - b. Desenhe o diagrama temporal da execução na maior frequência
- 2) (0,5) Mult.drs: $R1 = R2 \times R3$: Multiplicação de dois números com sinal ($-181 < R2 < 181$ e $-181 < R3 < 181$)
 - a. Filme o funcionamento
 - b. Desenhe o diagrama temporal da execução na maior frequência
- 3) (0,5) Divu.drs: $R1 = R2 / R3$: Divisão inteira de dois números sem sinal ($R2 < 65536$ e $R3 < 65536$)
 - a. Filme o funcionamento
 - b. Desenhe o diagrama temporal da execução na maior frequência
- 4) (0,5) Div.drs: $R1 = R2 / R3$: Divisão inteira de dois números com sinal ($-32768 < R2 < 32768$ e $-32768 < R3 < 32768$)
 - a. Filme o funcionamento
 - b. Desenhe o diagrama temporal da execução na maior frequência



- 5) (0,5) Remu.drs: $R1 = R2 \% R3$: Resto da divisão inteira de dois números sem sinal ($R2 < 65536$ e $R3 < 65536$)
 - a. Filme o funcionamento
 - b. Desenhe o diagrama temporal da execução na maior frequência
- 6) (0,5) Rem.drs: $R1 = R2 \% R3$: Resto da divisão inteira de dois números com sinal ($-32768 < R2 < 32768$ e $-32768 < R3 < 32768$)
 - c. Filme o funcionamento
 - d. Desenhe o diagrama temporal da execução na maior frequência
- 7) (1,0) Primo.drs: Apresente no display o primeiro número primo entre $R2$ e $R3$ ($0 < R2 < R3$).
 - e. Filme o funcionamento
 - f. Desenhe o diagrama temporal da execução na maior frequência
- 8) (1,0) Qual a maior frequência de clock que seu ZeptoProcessor-III consegue executar corretamente todos os programas anteriores? O que limita?

6. AVALIAÇÃO E CRONOGRAMA

Avaliação do protótipo será feita através do Relatório (2,0) e por um programa de Teste.drs (2,0), elaborado pelo professor e monitores, executado no seu processador.

No dia 05/11/2021 deve ser entregue pelo Aprender3 um arquivo com o nome 'Projeto_G04.zip' contendo todos os arquivos necessários para a execução do seu ZeptoProcessador-III no Deeds com o programa Primo.drs carregado na memória, programas (conteúdo da memória em arquivos .drs) e o .pdf do relatório no formato usual.

7. SUMÁRIO

Este projeto visa aplicar os conhecimentos adquiridos ao longo das disciplinas Circuitos Lógicos e Laboratório de Circuitos Lógicos para o desenvolvimento de um sistema digital programável, o ZeptoProcessador-III de 13 instruções.

8. EQUIPAMENTOS E MATERIAL

- Software Deeds