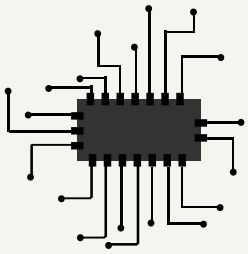


EMULADOR E ASSEMBLER

UNIVERSIDADE FEDERAL DO CEARÁ (UFC)

ARQUITETURA DE COMPUTADORES

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Descrição Geral do Trabalho

Projeção de uma microarquitetura, e implementação de seu respectivo emulador, permitindo que sejam executados programas quaisquer para ela escritos. Projeção de um conjunto de instruções para serem executadas pela microarquitetura, que será implementada pelo microprograma de controle. Por fim, implementação de um um montador (assembler) para uma linguagem de montagem (assembly) que traduza para a linguagem de máquina da macroarquitetura criada.

Emulador

Receberá arquivos binários com programas em linguagem de máquina da arquitetura e executá-los.
Sendo possível:
- Criar uma nova microarquitetura
- Melhorar a microarquitetura proposta em sala
- Manter a microarquitetura inalterada
- Criar um novo microprograma de controle, disponibilizando um novo conjunto de instruções (nova macroarquitetura)
- Melhorar o microprograma, tanto tornando mais eficientes as implementações das instruções já apresentadas, quanto adicionando novas instruções ao conjunto
- Manter o microprograma inalterado

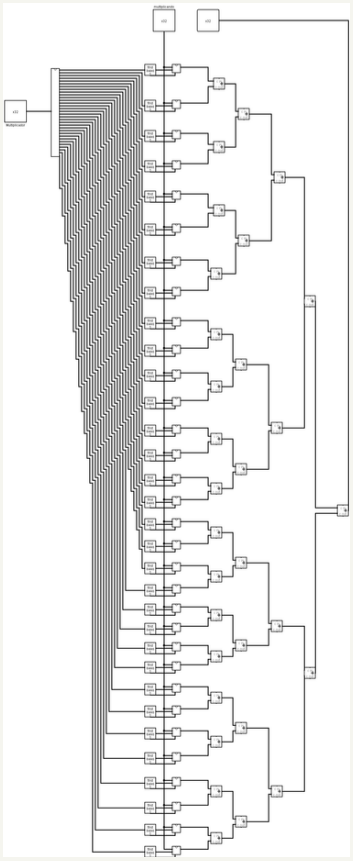
Assembler

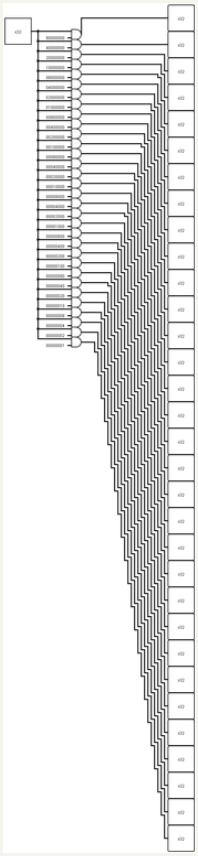
Deve montar programas escritos no assembly que será projetado na linguagem de máquina da arquitetura. O assembly deve cobrir todo o conjunto de instruções criadas na arquitetura bem como disponibilizar pseudo-instruções para escrita direta de bytes ou words no processo de montagem.

UNIVERSIDADE FEDERAL DO CEARÁ (UFC)
ARQUITETURA DE COMPUTADORES
Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Multiplicação (parte 1)

Explicação do circuito mulitplicador que foi adicionado à ALU





O multiplicador consiste, basicamente, na soma dos produtos parciais de um número inteiro por outro.

Ao obter os produtos parciais por meio do subcircuito extensor, temos 32 multiplicações a serem feitas entre (multiplicador) x (uma potência de dois encontrada ou um valor nulo). Dessa forma, usamos o circuito indexador que recebe como entrada uma potência de dois que existe no multiplicando.
Assim, o indexador apresenta duas saídas:
1 - 5 bits que representam a posição do bit 1 na potência de dois (saída assume valor nulo se não houver bit 1).
2 - 1 bit que representa um booleano se o bit 1 foi encontrado ou não (1 para encontrado, 0 para não encontrado). A saída 2 é muito importante, visto que a saída 1 pode ser nula tanto quando o bit 1 não for encontrado, tanto quando o bit 1 estiver localizado no primeiro bit (índice 0).

Assim, teremos a posição de cada bit 1 de cada potência de dois do multiplicador, e usamos o valor da posição para calcular a multiplicação entre o multiplicando e cada potência de dois (obtenção dos produtos parciais), para assim somarmos todos esses produtos e obtermos o valor da multiplicação dos dois números entrados.

SUBCIRCUITO EXTENSOR
(cálculo dos produtos parciais)

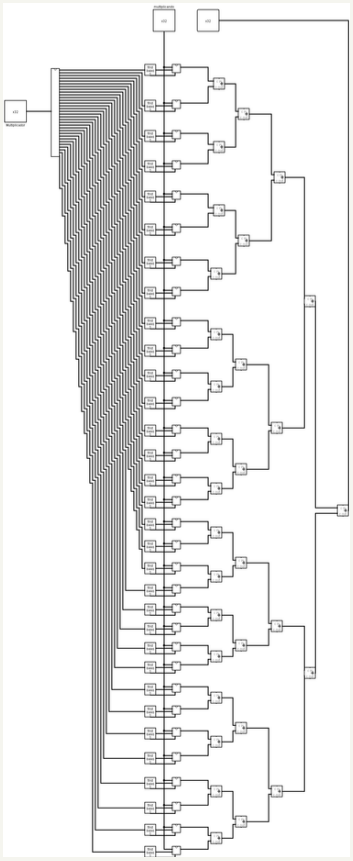
Por meio de operações de comparações usando portas AND do multiplicador, auxiliadas de constantes (cada constante assume o valor de uma potência de dois pertencente ao conjunto {2^0, 2^1, ..., 2^31}), obteremos 32 saídas de 32 bits.

Cada saída contém uma potência de dois do multiplicando, ou um valor nulo. Portanto, com esse circuito, temos todas as potências de dois do "multiplicando".

UNIVERSIDADE FEDERAL DO CEARÁ (UFC)
ARQUITETURA DE COMPUTADORES
Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

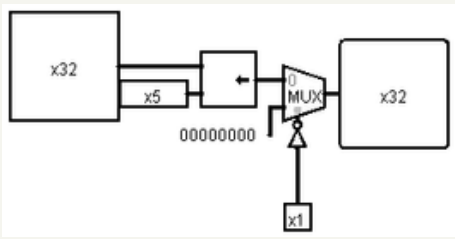
Multiplicação (parte 2)

Explicação do circuito mulitplicador que foi adicionado à ALU



Com o auxílio do indexador e do deslocador, temos todas as multiplicações parciais, restando apenas somar tais valores para que assim seja obtido como resultado a multiplicação em si.
Para realizar a soma, usamos circuitos somadores que vão somando dois valores por vez dessa forma, pegamos o resultado da soma e usamos como entrada para outras somas, até que todos os valores sejam somados e o resultado seja obtido.

Obs: a multiplicação será feita em um ciclo de clock.



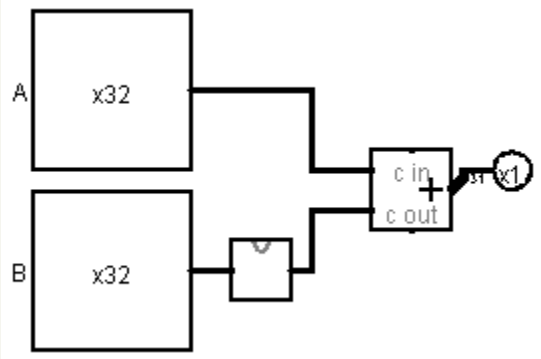
SUBCIRCUITO DESLOCADOR
(realiza o produto parcial)

O circuito deslocador receberá três entradas:
1 - o multiplicando
2 - o valor que representa a quantidade de bits que serão deslocados.
3 - um bit que representa se há um valor 1 na potência de dois obtida anteriormente ou não.
Assim poderemos fazer, através de deslocamentos para a esquerda, a multiplicação de uma potência de dois do multiplicador pelo multiplicando, ou, se não for encontrado nenhum bit 1 na potência de dois, zeramos a saída através de um multiplexador.

UNIVERSIDADE FEDERAL DO CEARÁ (UFC)
ARQUITETURA DE COMPUTADORES
Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

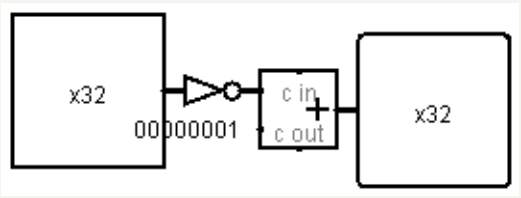
Comparação - Menor que

Explicação do circuito "menor que" que foi adicionado à ALU



O "menor que" consiste, basicamente, na ação de fazer a soma de uma entrada A da ALU, com uma entrada B da ALU invertida e somada com 1.
Ao inverter o valor de B e somar com A, teremos uma subtração de A por um valor B' , que resultará em um número (pegamos o bit mais à esquerda desse número a fim de saber o sinal do resultado) positivo (saída 0) ou negativo (saída 1), indicando qual número é menor.

Primeiramente, a entrada A recebida já é direcionada para o circuito somador, enquanto a entrada B vai ser tratada antes de ser direcionada ao encontro de A.



SUBCIRCUITO TRATAMENTO DE B
(INVERTE E SOMA 1)

No tratamento de B, invertemos o número utilizando a porta NOT conectada ao número. Após isso, somamos o número 1 (em binário de 32 bits) ao resultado gerado pela inversão.

Dessa forma, ao fim do tratamento de B, o mesmo é somado com o valor recebido na entrada A, gerando uma saída 0 para MAIOR OU IGUAL QUE, ou uma saída 1 para MENOR QUE.

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Organização da WORD

9 bits — de endereçamento

3 bits — de salto

8 bits — da ALU

8 bits — de registradores para C

antes o número total era de 6 registradores, mas em nossa microarquitetura optou-se para utilização de 8

3 bits — de memória

3 bits — do barramento B

3 bits — do barramento A

foi acrescentado 3 bits para o tratamento de valores direcionados ao barramento A

37 bits

logo, esse é o total de bits usados para a word

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Bits de endereçamento

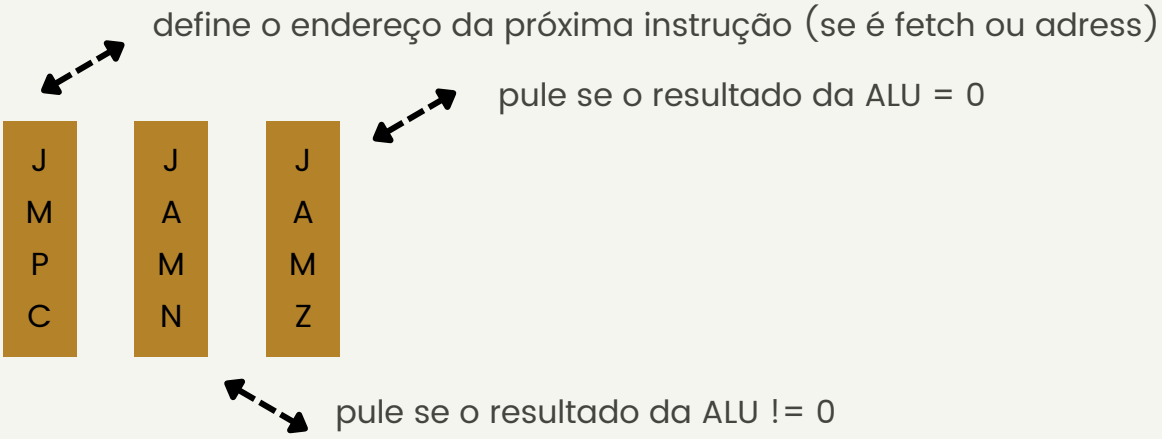
9 bits de endereçamento

Utilizamos de 9 bits para essa função, que terá como principal funcionalidade determinar o endereço da próxima instrução a ser executada.

Bits de salto

3 bits de salto

Utilizamos de 3 bits para essa função, que terá como principal funcionalidade determinar se o endereço da próxima instrução a ser executada. vem do endereçamento ou do MBR. Além de determinar a ação de salto de acordo com o resultado calculado pela ALU



instruções adicionais

111	vai pra main se o resultado da ALU != 0
110	vai pra main se o resultado da ALU = 0

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Bits da ALU

8 bits da ALU

Utilizamos de 8 bits para determinar o papel que será desempenhado pela ALU, a qual realiza operações lógicas e aritméticas sobre os dados. Todos os outros elementos do sistema trazem dados para a ALU processar, e depois levar os resultados de volta.

Comandos da ALU:

Possíveis combinações de bits e suas respectivas funcionalidades:

011000	A	110110	B-1
010100	B	111011	-A
011010	~A	001100	A e B
101100	~B	011100	A ou B
111100	A+B	010000	0
111101	A+B+1	110001	1
111001	A+1	110010	-1
110101	B+1	001101	A x B
111111	B-A	011101	A<B

Foram acrescentados, por meio da criação de novos circuitos, as funções de "Multiplicação" e "Menor que", que serão apresentadas a seguir

Bits de C

-------- de C

8 bits

Utilizamos de 8 bits para essa função, que terá como principal funcionalidade utilizar uma sequência de bits para determinar qual registrador será gravado no barramento C

Possíveis combinações de bits e suas respectivas funcionalidades:

10000000	MAR
01000000	MDR
00100000	PC
00010000	X
00001000	Y
00000100	W
00000010	Z
00000001	H

Bits de Memória

--- de memória

3 bits

Utilizamos de 3 bits para essa função com o objetivo de fazer a manipulação da memória, e acionar o papel de escrita e leitura na memória

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Bits do barramento B

3 bits — do barramento B

Utilizamos de 3 bits para determinar qual o registrador que será lido para ter seu valor direcionado ao barramento B

000	MDR
001	PC
010	MBR
011	X
100	Y
101	W
110	Z
111	H

Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Bits do barramento A

3 bits — do barramento A

Utilizamos de 3 bits para determinar qual o registrador que será lido para ter seu valor direcionado ao barramento A

000	MDR
001	PC
010	MBR
011	X
100	Y
101	W
110	Z
111	H

UNIVERSIDADE FEDERAL DO CEARÁ (UFC)
ARQUITETURA DE COMPUTADORES
Alunos:
Gabriel Alves - 511318
Thomaz Ângelo - 509299

Documentação do Firmware

Microinstruções de função específica que estão alocadas em um endereço de memória

FUNÇÃO	ENDEREÇO	MNEMÔNICO
halt	255	halt
X = X + mem[adress]	2	add x,
mem[adress] = X	5	mov x,
GOTO adress	8	goto
X = X - mem[adress]	10	sub x,
X = X * mem[adress]	13	mult x,
if X == 0 then GOTO adress	16	jz x,
X = X < mem[address]	18	smlr x,
X = X //mem[address]	21	div x,
X = X % mem[address]	26	mod x,
X = min(X, mem[address])	29	min x,
X = X == mem[address]	33	ig x,
X = X > mem[address]	37	bigr x,
X = X >> 1	40	desl1
X = X >> 2	41	desl2