Optativa - ARQUITETURA DE COMPUTADORES

Aluno: Daniel Sant' Anna Andrade
Matrícula: 20200036904

1) O pipeline de instrução tem como objetivo executar instruções simultaneamente. Quando uma instrução executa o ciclo de busca e passa para o ciclo de decodificação, o ciclo de busca para de ser necessário para a primeira instrução e fica disponível para executar outra instrução. Isso irá ocorrer durante todo o pipeline, podendo executar até 5 instruções diferentes ao mesmo tempo, caso não ocorra dependência de dados.

Exemplo: o tempo total gasto para executar um conjunto de 5 instruções lw sem o pipeline iria ser de 40ns. Agora usando o pipeline para executar o mesmo conjunto (caso não ocorra dependência de dados) o tempo total será de 17ns.

2) O mapeamento associativo por conjunto é semelhante ao mapeamento direto, organizando os blocos de memória pelo índice dos endereços. Porém, no associativo por conjunto, teremos diferentes rótulos associados a um único index, o que não ocorre na por mapeamento direto, diminuindo assim as chances de conflitos e aumentando a taxa de acerto.

5)

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
ADD R3 R11 R12	BI	REG	EXE	MEM	[redacted]					
LW R3 30($8)		BI	REG	EXE	MEM	[redacted]				
OR R4 R2 R5			BI	REG	EXE	MEM	[redacted]			
AND R7 R6 R4				BI	[redacted]	EXE	MEM	REG		
LW R6 10($9)					BI	[redacted]	EXE	MEM	REG	
SUB R7 R5 R8						BI	[redacted]	EXE	MEM	REG

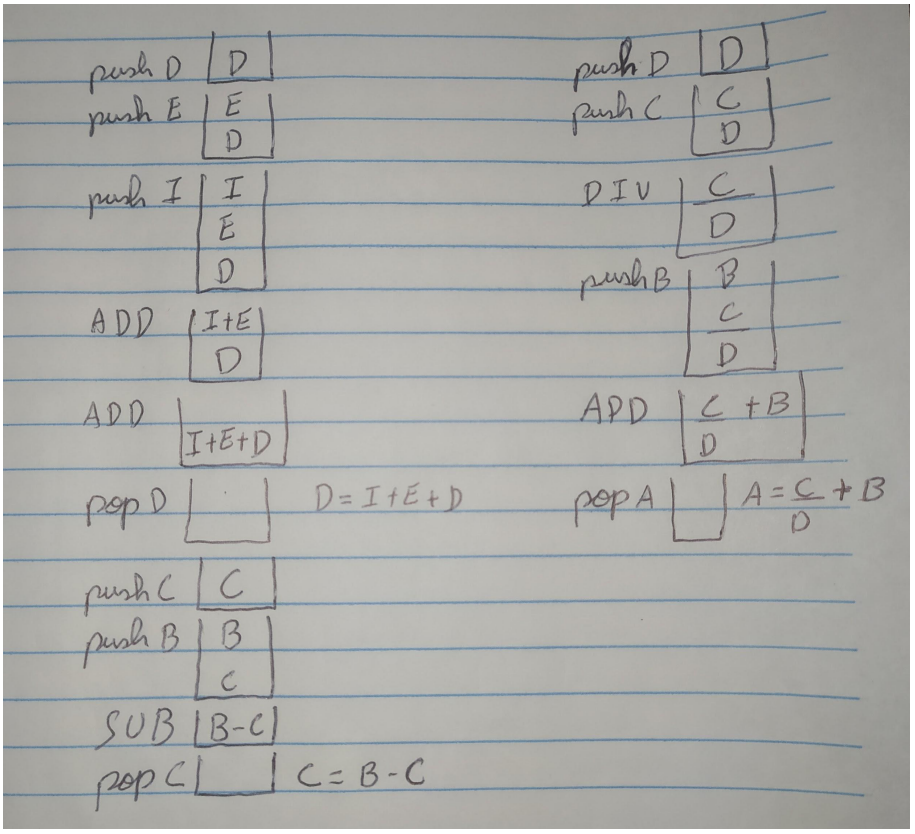
Foram utilizados 10 ciclos para executar o código. Apesar de ocorrer dependência de dados entre a terceira e quarta instrução, como o pipeline faz adiantamento de dados, o valor de R4 estará pronto depois do ciclo de execução, sendo mandado diretamente para o início do ciclo de execução da quarta instrução através dos registradores do pipeline. Já nos ciclos T5, T6 e T7 apesar dos registradores estarem sendo utilizados mais de uma vez isso é possível pois cada um só utiliza 1ns do ciclo para executar, com o ciclo de escrita sendo executado antes do ciclo de leitura.

6)
( ) No computador MIPS, admitindo-se a duração de 2ns por estágio do pipeline, a leitura dos registradores é feita no primeiro 1ns do 2º estágio.
(x) Somente as instruções de load (lw) e Store (Sw) fazem algo de útil no 4 o estágio do MIPS.
(x) A predição de desvio fixa tem um desempenho melhor do que a predição dinâmica de desvio que utiliza apenas um bit.
(x) No mapeamento completamente associativo que faz uso da localidade espacial, o conteúdo da memória cache é composto por: endereço completo e o dado informação).
(x) Uma característica do modelo de cache completamente associativo é a utilização ao máximo da capacidade da cache.
( ) O principal algoritmo utilizado para substituição de blocos na cache com modelo de mapeamento direto é o LRU.
( ) No modelo associativo por conjunto, a parte dos bits mais significativo do endereço é armazenado na cache no campo chamado de índice.

( ) No mapeamento completamente associativo cada bloco da memória principal só pode ser mapeado em uma única posição da memória cache.
(x) Uma característica da arquitetura RISC é a unidade de controle microprogramada.

7) No princípio da localidade temporal, temos como princípio que ao utilizar uma instrução da memória, a chance de precisar utilizar novamente essa instrução é alta, por conta disso, é necessário armazenar dados que são muito utilizados em uma memória mais rápida - como a memória cache - para ter mais velocidade no momento que precisar buscar novamente essa informação.

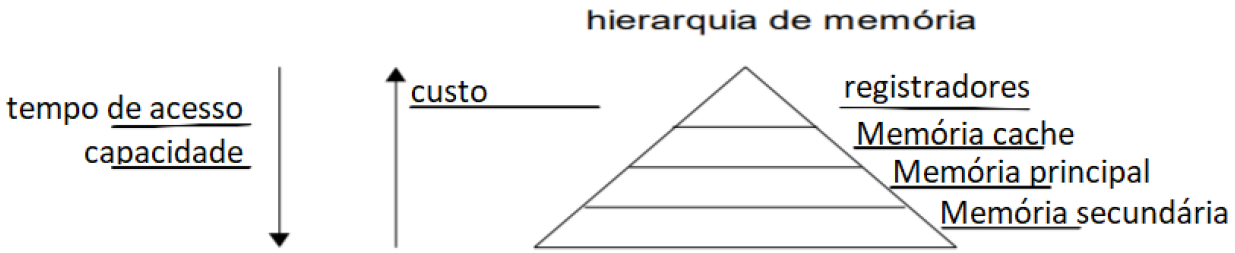
8)



9) O LRU (least recently used) é um método de substituição usado quando a cache de mapeamento completamente associativo ou de mapeamento associativo por conjunto está lotada. Esse algoritmo substitui o bloco da cache que está a mais tempo sem ser usado. São utilizados bits associados às posições da cache para marcar o tempo em que um dado foi alocado na mesma. Ao precisar substituir um valor, quando a cache estiver cheia, é visto esse bit de tempo para que possa ser feita a escolha da informação que será retirada.

10)
a) -1
b) -0
c) -127

11)



12)
Dividindo em partes:

11 em binário é 1011
0,5625 em binário é 0.1001
Logo, 11,5625 é 1011.1001

Esse resultado no padrão IEEE é $1011.1001 \cdot 2^{0}$
Movendo a vírgula para a direita, fica: $1.0111001 \cdot 2^{3}$

Na tabela, é necessário primeiramente descobrir o expoente, somando o 3 com 127.

128	64	32	16	8	4	2	1	
	1	1	1	1	1	1	1	127
				0	0	1	1	3
1	0	0	0	0	0	1	0	130

Agora é só alocar os valores na tabela de 32 bits, o bit mais significativo (31) é o bit que indica o sinal, caso seja positivo o bit é 0, caso seja negativo o bit é 1. No caso dessa questão será 1.

Do 30 ao 23, se coloca o expoente em binário (a soma do expoente achado com 127, no meu caso 130).

O restante será a mantissa, o valor depois da vírgula após encontrar o expoente no caso 0111001. Os espaços que sobrarem se completa com 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C				1				3				9				0				0				0				0			

Em hexadecimal o valor de -11,5625 convertido para o formato IEEE 754 de precisão simples é C1390000.