Número: _____ Nome: _____

1. Considere a seguinte sequência de código assembly do Y86:

Assembly	**Binário**

`.pos 0x010B0`
`irmovl $10, %eax`
`pushl %eax`
`call func`
`popl %eax`
`halt`
`.pos 0x02000`
`func:`
`ret`	

a) Preencha a coluna da direita do quadro anterior com o código máquina correspondente, indicando os endereços iniciais de cada instrução.

b) Para a instrução "`irmovl $10, %eax`", preencha a tabela seguinte com os valores relevantes dos sinais do *datapath*, para a versão SEQ (sequencial) do processador Y86.

Extracção		Memória	
Descodificação		Actualização	
Execução			

c) Para a instrução "`ret`", preencha a tabela seguinte com os valores relevantes dos sinais do *datapath*, para a versão SEQ (sequencial) do processador Y86.
NOTA: suponha que o `%esp` tem o valor 0x0A000 no início da execução desta instrução.

Extracção		Memória	
Descodificação		Actualização	
Execução			

d) Indique, justificando sucintamente, o valor do PC (Program Counter) no início do ciclo 4, para as organizações PIPE- e PIPE do Y86.

Justificação:

Número: _____ Nome: _____

2. Considere que o conjunto de instruções do Y86 é aumentado com a instrução de adição condicional

```
addlxx rA, rB
```

Esta instrução soma os conteúdos dos registos rA e rB, escrevendo o resultado em rB se e só se a condição xx for verdadeira. O conjunto de condições possíveis é o mesmo que para os saltos condicionais.

A sua realização na organização PIPE, estágio de execução, ocorre da seguinte forma:

- a ALU soma os valores valA e valB;
- a condição é avaliada, tomando como entradas os valores dos códigos de condição anteriores à execução desta instrução;
- no fim do ciclo:
  - M_valE é actualizado com o resultado da adição;
  - Se a condição for verdadeira:
    - M_dstE é actualizado com o valor de E_dstE;
    - Os códigos de condição são actualizados de acordo com o resultado da adição;
  - Se a condição for falsa:
    - M_dstE é actualizado com o valor 8, indicando que o resultado não deve ser escrito em nenhum registo na fase de actualização;
    - Os códigos de condição não são alterados;

Indique, justificando, se os atalhos da organização PIPE existentes para resolver dependências de dados processados na ALU devem/podem ser utilizados em cada um dos seguintes casos:

Caso	Justificação

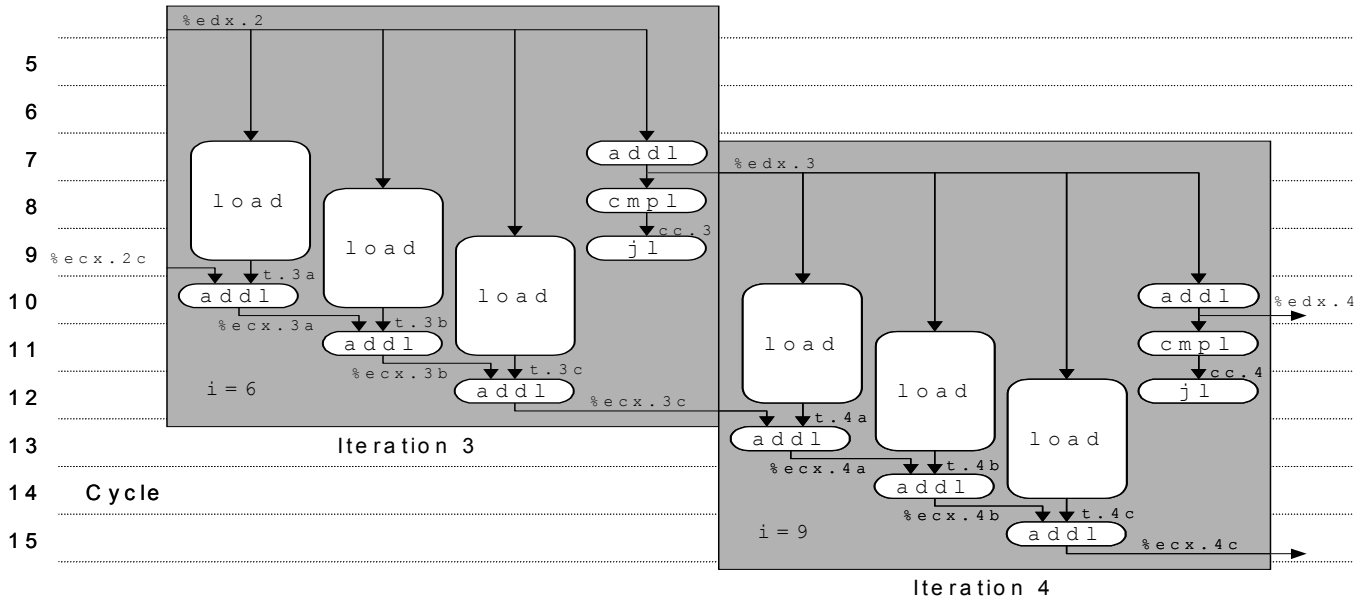
Atalho: W_valE
`addlxx %eax, %ebx`
`nop`
`nop`
`addl %ebx, %ebx`	

Atalho: M_valE
`addlxx %eax, %ebx`
`nop`
`addl %ebx, %ebx`	

Atalho: e_valE
`addlxx %eax, %ebx`
`addl %ebx, %ebx`	

Número: _____ Nome: _____

3. A figura representa o diagrama de ocupação das unidades funcionais de um processador baseado na arquitectura IA32.



a) Quais as **características** temporais das diferentes unidades funcionais representadas e qual o valor do **CPE**?

b) Caracterize a diagrama de execução quanto à quantidade de recurso usados (**limitado/ilimitado**). Justifique!

c) Apresente o corpo básico de **instruções elementare**s correspondente ao fragmento de código que descreve cada uma das iterações.

Número: _____ Nome: _____

d) Que **técnica** de optimização de desempenho está implícita no diagrama e que **objectivos** prossegue? Justifique!

4. – Apresente sucintamente o postulado conhecido por lei de More, e as consequências resultantes da sua verificação, ao longo das últimas décadas.

5. Caracterize a abordagem de tratamento de interrupções implícita na figura. Justifique detalhadamente.