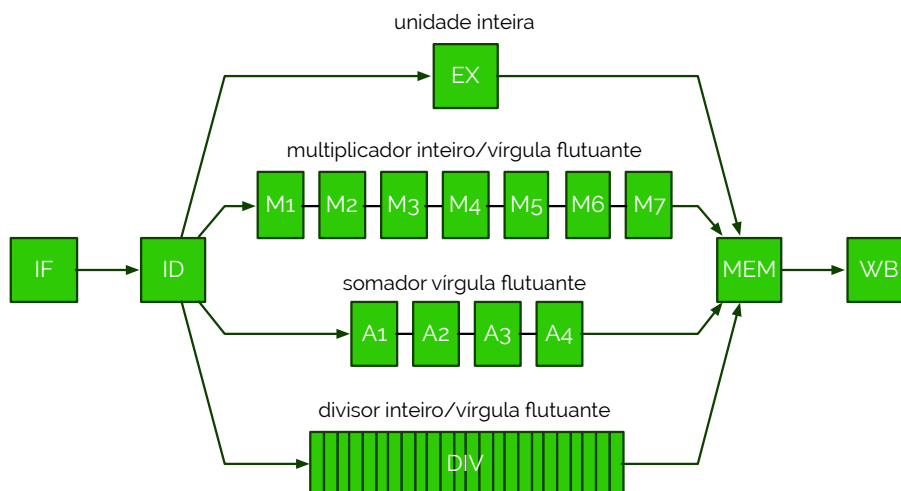


Esta ficha foi elaborada para acompanhar o estudo da disciplina de Arquitetura de Computadores Avançada (a4s1) ou para complementar a preparação para os momentos de avaliação finais da mesma. Num segundo ficheiro poderás encontrar um conjunto de propostas de solução aos exercícios que estão nesta ficha. É conveniente relembrar que algum conteúdo destes documentos pode conter erros, aos quais se pede que sejam notificados pelas vias indicadas na página web, e que serão prontamente corrigidos, com indicações de novas versões.

1. Explica no que consiste o esquema de delayed branch (branch em atraso).
2. O que é uma exceção? Dá exemplos de 2 exceções que ocorrem no curso (within) e entre (between) instruções. Identifica 2 exceções para cada tipo.
3. "As exceções são atendidas por uma rotina de tratamento que resolve os problemas que condicionaram a execução normal do programa e restaura o mesmo para o momento antes do seu lançamento". Concordas com a afirmação, para qualquer caso de exceção?
4. As exceções podem ter um variado leque de causas para o seu lançamento, contudo, elas podem ser agrupadas em vários conjuntos diferentes que definem o seu tipo/a sua categoria. Dos cinco tipos seguintes, explica sucintamente no que se baseiam e dá um dois casos de aplicação (para uma categoria definida pelo par A, B dá um exemplo para A e um exemplo para B):
 - a) Exceções síncronas e assíncronas;
 - b) Exceções maskable e non-maskable;
 - c) Exceções within e between (se respondeste à questão 2 apenas precisas de explicar o conceito);
 - d) Exceções resume e terminate;
 - e) Exceções pedidas pelo utilizador e coagidas.
 - f) Uma exceção pode fazer parte de duas ou mais categorias, ou é exclusiva a uma só, das referidas? Se puder fazer parte de duas ou mais categorias dá exemplo de duas exceções e categoriza-as com o máximo de tipos possível.
5. O que é que significam "exceções precisas"? Porque é que muitos dos processadores atuais introduziram dois modos de operação: um com exceções precisas e outro sem?
6. Consideremos a figura abaixo onde mostramos um pipeline clássico de cinco andares com uma pequena variante de operações multi-cycle na fase de execução.



- a) Explica de que forma é que a execução das instruções neste pipeline podem ser descritas como em-ordem com, eventualmente, out-of-order completion (conclusão fora-de-ordem).
- b) Que tipos de hazards estruturais podem ocorrer neste pipeline?
- c) Que tipos de hazards de dados podem ocorrer neste pipeline?

d) Considera o seguinte código a ser executado neste pipeline:

```
div.d  f0, f2, f4
add.d  f10, f10, f8
sub.d  f12, f12, f14
```

É possível a operação de subtração causar uma exceção aritmética de vírgula flutuante em tal ponto que a instrução de soma já terminou a execução, mas a operação de divisão ainda não? Em qualquer dos casos, numa situação semelhante, supõe que a operação de divisão lançava uma exceção antes da operação de soma terminar. Classifica essa exceção como precisa ou imprecisa, justificando.

7. "Qualquer dependência entre instruções inibe estas de serem executadas em paralelo". Concordas com a afirmação?

8. Distingue antidependências de dependências de saída.

9. Dá um caso de aplicação (código ou figura) onde o hazard de dados do tipo RAR está presente.

10. Consideremos as instruções A e B, com A a preceder B no programa. "O hazard de dados WAR ocorre quando A tenta escrever um valor num operando antes que B o leia, pelo que a instrução B irá obter um valor errado". Concordas com a afirmação?

11. Porque é que o hazard do tipo WAR não poderá ocorrer em grande parte dos pipelines?

12. A execução fora-de-ordem é uma funcionalidade do agendamento dinâmico. De forma a implementá-lo, a fase de instruction decode do pipeline clássico de cinco andares deve ser decomposto em duas sub-fases. Que fases são essas e qual é o papel delas?

13. Duas técnicas básicas para implementação do agendamento dinâmico são o scoreboarding e o algoritmo de Tomasulo. Qual é a diferença substancial entre ambas?

14. Considera a seguinte sequência de instruções, a ser executado num pipeline onde o agendamento dinâmico é feito por scoreboarding, tendo duas unidades funcionais de soma e duas unidades funcionais de multiplicação (ambas em vírgula flutuante):

```
add.d  f0, f2, f4
mult.d f2, f6, f8
mult.d f10, f0, f2
add.d  f0, f12, f14
```

estado de instrução				
instrução	lançamento	leitura de operandos	execução	escrita de resultado
add.d f0, f2, f4				
mul.d f2, f6, f8				
mul.d f10, f0, f2				
add.d f0, f12, f14				

estado de unidades funcionais									
nome	busy	op	F _i	F _j	F _k	Q _j	Q _k	R _j	R _k
mult1	não								
mult2	não								
soma1	não								
soma2	não								

estado de registos								
F0	F2	F4	F6	F8	F10	F12	...	F30

a) Completa a tabela acima considerando que apenas foi lançada a instrução add.d f0, f2, f4.

b) Completa a tabela acima considerando que a instrução add.d f0, f2, f4 terminou a escrita de resultados.

c) Completa a tabela no ciclo seguinte ao da alínea b).

d) Considerando o primeiro ciclo de execução em $t = 0$, em que ciclo de execução é que a alínea c) está?

15. Considera novamente o código do exercício 14 (repetido em baixo) a ser executado num pipeline onde o agendamento dinâmico é feito, agora, pelo algoritmo de Tomasulo, tendo duas unidades funcionais de soma e duas unidades funcionais de multiplicação (ambas em vírgula flutuante):

```
add.d f0, f2, f4
mult.d f2, f6, f8
mult.d f10, f0, f2
add.d f0, f12, f14
```

estado de instrução (não faz parte do hardware)			
instrução	lançamento	execução	escrita de resultado
add.d f0, f2, f4			
mul.d f2, f6, f8			
mul.d f10, f0, f2			
add.d f0, f12, f14			

estações de reserva / buffers							
nome	busy	op	V_j	V_k	Q_j	Q_k	A
mult1	não						
mult2	não						
soma1	não						
soma2	não						

estado de registos									
Campo	F0	F2	F4	F6	F8	F10	F12	...	F30
Q_i									

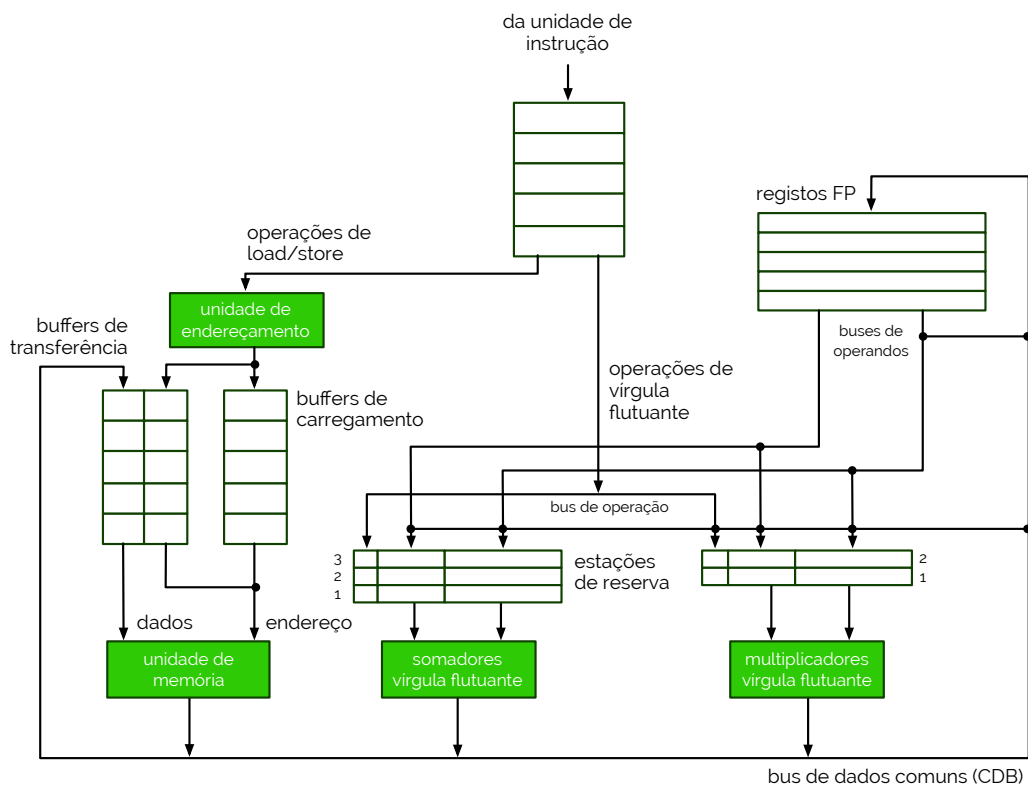
a) Completa a tabela acima considerando que apenas foi lançada a instrução add.d f0, f2, f4.

b) Completa a tabela acima considerando que a instrução add.d f0, f2, f4 terminou a escrita de resultados no bus de dados comum (CDB).

c) Completa a tabela no ciclo seguinte ao da alínea b), considerando que a multiplicação tem uma latência maior que 1.

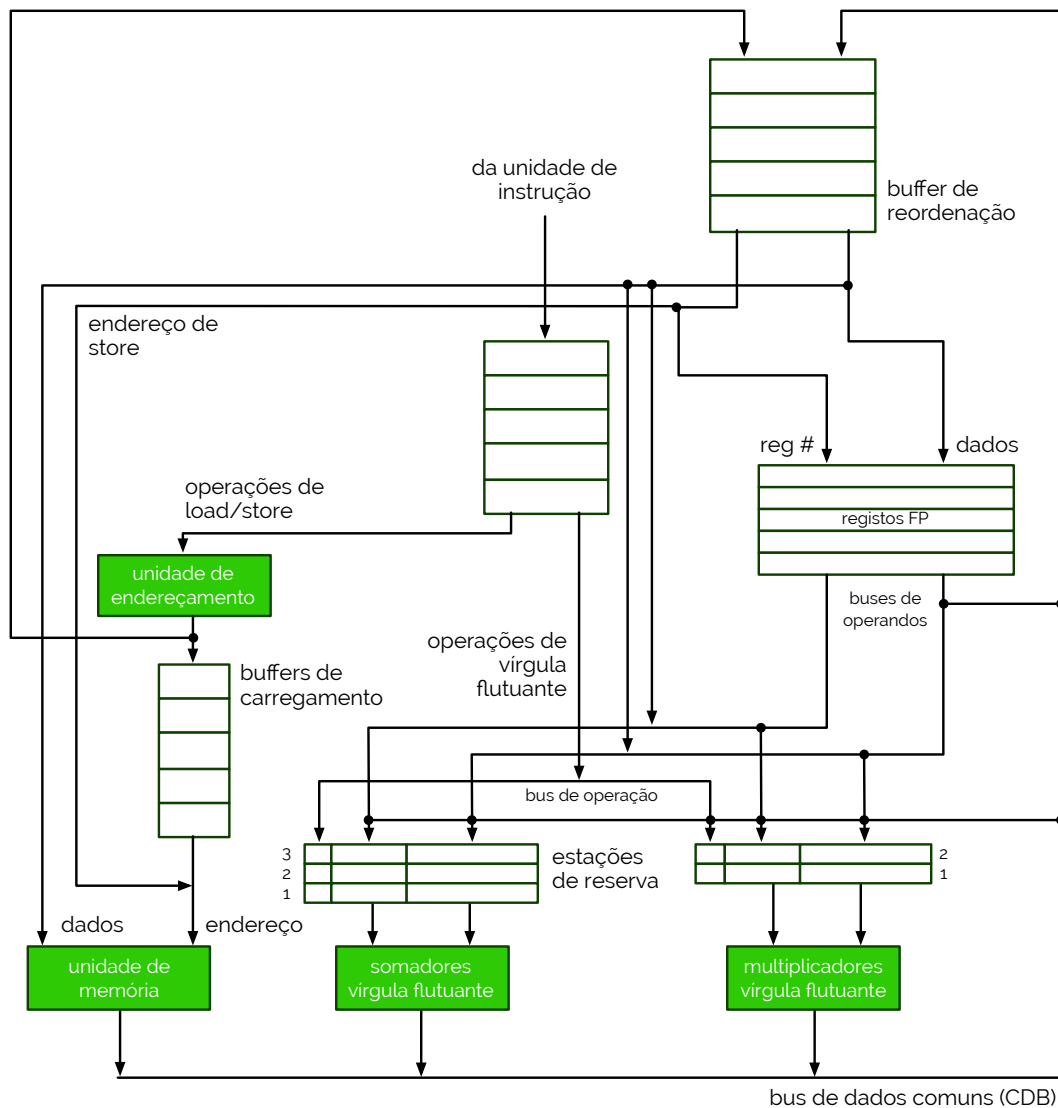
d) Considerando o primeiro ciclo de execução em $t = 0$, em que ciclo de execução é que a alínea c) está?

16. Considera a figura abaixo, onde se exhibe o conceito de algoritmo de Tomasulo.



- Quantos registros-resultado estão presentes na figura acima?
- O que são os registros-resultado?
- "As instruções, até à sua execução, estão em-ordem, com o algoritmo de Tomasulo". Concordas com a afirmação?
- Como é que as instruções são identificadas nas estações de reserva? Qual é a importância da sua identificação?
- Se o algoritmo de Tomasulo se baseia na correção de dependências de nome, como é que resolve os hazards do tipo RAW?

17. Considera a figura abaixo.



- a) Que aplicação de agendamento dinâmico é que está a ser representada na figura acima? Justifica a tua resposta.
- b) O que é que foi introduzido com esta implementação que o algoritmo de Tomasulo não assumia?
- c) O que é o "buffer de reordenação" visível na figura? Qual foi o intuito da sua criação?

18. Considera o código seguinte, a ser executado num pipeline com especulação por hardware, considerando também a figura do exercício 17.

```
l.d    f6, 34(r2)
l.d    f2, 45(r3)
mul.d  f0, f2, f4
sub.d  f8, f6, f2
div.d  f10, f0, f6
add.d  f6, f8, f2
```

buffer de reordenação						
entrada	estado	busy	instrução	destino	value	ready
1		não				
2		não				
3		não				
4		não				
5		não				
6		não				

estações de reserva / buffers								
nome	busy	op	V _j	V _k	Q _j	Q _k	destino	A
load1	não							
load2	não							
add1	não							
add2	não							
add3	não							
mult1	não							
mult2	não							

estado de registos									
Campo	F0	F2	F4	F6	F8	F10	F12	...	F30
busy									
ent. ROB									

a) Completa a tabela acima considerando que a primeira instrução foi lançada.

Considera a partir deste ponto, as seguintes latências: operações de soma e load/store - 2 ciclos; operações de multiplicação - 10 ciclos; operações de divisão - 40 ciclos.

b) Completa a tabela considerando que a operação de multiplicação foi lançada.

c) Completa a tabela acima considerando o ciclo seguinte à alínea b).

d) Completa a tabela considerando que é feito o commit da primeira instrução.

e) Completa a tabela considerando que a última instrução foi lançada.