

Arquitetura de Computadores

1º Trabalho de Laboratório Unidade de Processamento

Objetivo: pretende-se que os alunos compreendam a metodologia usada na síntese, implementação e teste de uma Unidade de Processamento elementar. O trabalho terá uma duração de 2 semanas. O relatório final será entregue até à sexta-feira da segunda semana de laboratório, ou seja, **até dia 24/3/2017**, via Fénix, em formato “pdf” juntamente com o projeto do Vivado. O trabalho deverá ser preparado (projeto, edição dos esquemas e testes) fora do horário de laboratório, destinando-se as 3 horas de laboratório à resolução de eventuais problemas e demonstração do trabalho realizado.

1 INTRODUÇÃO

Para a realização do presente trabalho é disponibilizada uma Unidade de Processamento composta por uma ALU, uma unidade de armazenamento com registos, memória e multiplexers. A unidade está completamente implementada e testada, e encontra-se no ficheiro DP.zip que se encontra na página da cadeira.

2 ANÁLISE E CARACTERIZAÇÃO DE UMA UNIDADE DE PROCESSAMENTO (1ª SEMANA)

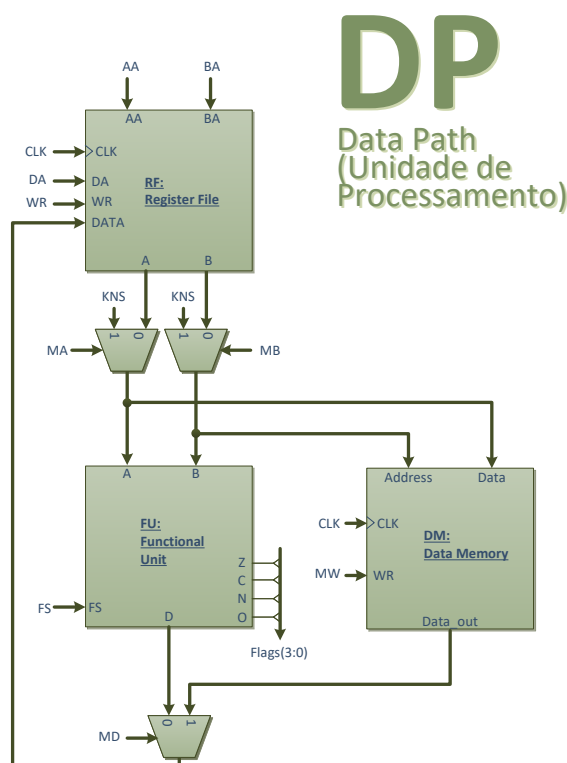


Figura 1 – Diagrama de blocos da Unidade de Processamento fornecida.

Nesta secção pretende-se fazer a análise e caracterização, da Unidade de Processamento fornecida. O estudo desta unidade deverá ser suficientemente detalhado para facilitar a posterior alteração da sua arquitetura no sentido de incluir diferentes funcionalidades.

Para analisar a unidade de processamento deve estudar tanto o código VHDL fornecido como os esquemas elaborados pelo Vivado. Note que o projeto foi desenvolvido de uma forma que os esquemas gerados sejam de fácil leitura e que estes devem servir de ajuda a compreender o código VHDL. **Abra e estude todos os ficheiros de código fonte fornecidos.** Estes ficheiros devem servir de exemplo para as modificações requeridas de futuro.

1. **[4 Val.]** Caracterize completamente a unidade funcional (functional unit, FU), identificando:
 - a) As operações que a unidade é capaz de realizar através da análise dos esquemas elaborados pelo Vivado e dos ficheiros VHDL fornecidos. Para cada operação identificada indique o estado do sinal FS e a função na saída numa tabela. Inclua no relatório uma justificação sucinta de como chegou a estas conclusões. No caso da unidade aritmética faça uma tabela de verdade da lógica de entrada B de cada somador completo (full-adder) e da entrada de transporte (carry in) do somador.
 - b) Quais as flags válidas para cada uma das operações identificadas. Justifique sucintamente.
2. **[2 Val.]** Identifique e caracterize a Unidade de Armazenamento através da análise dos esquemas dos ficheiros VHDL. Inclua no relatório uma tabela com a descrição e função de cada um dos sinais da unidade de armazenamento.
3. **[3 Val.]** Faça um teste do funcionamento da unidade de processamento de forma a realizar as operações indicadas na tabela seguinte; para cada operação indique numa tabela a palavra de controlo respetiva, i.e., o valor de todos os sinais de controlo necessários à execução da operação. Faça uma única simulação com os diferentes micro operações em sequência, e com uma micro operação por ciclo de relógio. Apresente ainda na tabela os resultados esperados e compare com o resultado da simulação. Siga o exemplo apresentado na Tabela 2. Para realizar as simulações modifique o ficheiro SimulDataPath.vhd fornecido junto com o projeto.

Tabela 1 – Simulações a realizar por dia da semana

(Turnos de 2ª feira)	(Turnos de 3ª feira)	(Turnos de 4ª feira)	(Turnos de 5ª feira)	(Turnos de 6ª feira)
R1 ← 2	R1 ← 3	R1 ← 4	R1 ← 5	R1 ← 6
R2 ← <Número formado pelos 4 dígitos menos significativos do número de aluno do 1º elemento do grupo>				
R3 ← <Número formado pelos 4 dígitos menos significativos do número de aluno do 2º elemento do grupo>				
R4 ← LSL R3	R4 ← ROR R3	R4 ← LSR R3	R4 ← ROL R3	R4 ← ROR R3
R4 ← R2 + R4	R4 ← R4 + 10	R4 ← R2 – R4	R4 ← – R4	R4 ← R4 + 5
R4 ← NOT R4	R4 ← R4 XOR R1	R4 ← R4 AND R3	R4 ← R4 OR R2	R4 ← R4 XOR R2
R1 ← R4 – 7	R3 ← R1 + 7	R1 ← R4 – 14	R1 ← R2 – 3	R3 ← R1 – 25

Tabela 2 – Exemplo de tabela com as palavras de controlo

Sinais de controlo										V. Esperado		Resultado	
AA	BA	DA	WR	MA	MB	KNS	FSUF	MD	MW	D	Flags	D	Flags

4. **[1 Val.]** Considere a realização da 5ª operação da tabela anterior. Verifique na simulação qual o valor do sinal D(31:0) à saída da FU antes e depois do flanco de relógio. Comente o que observa.

3 SÍNTESE DE NOVAS FUNCIONALIDADES PARA A UNIDADE DE PROCESSAMENTO (2ª SEMANA)

[5 + 5 Val.] Considere a unidade de processamento fornecida e proceda ao menor número de alterações para que esta realize todas as funções indicadas nas perguntas seguintes deixando as restantes funcionalidades inalteradas em relação à arquitetura original.

Faça as alterações referidas Tabela 3 para o seu turno. A descrição detalhada das alterações está de seguida.

Inclua no relatório: 1) código VHDL com as alterações efetuadas; 2) os esquemas gerados pelo Vivado; 3) o resultado das simulações; 4) o projeto do Vivado.

Tabela 3 – Alterações a realizar por dia da semana

(Turnos de 2ª feira)	(Turnos de 3ª feira)	(Turnos de 4ª feira)	(Turnos de 5ª feira)	(Turnos de 6ª feira)
Alterar entradas da memória.	Alterar a unidade aritmética 1.	Transferências entre registos no RF.	Acrescentar um multiplicador à unidade funcional.	Alterar a unidade aritmética 2.
Alterar a unidade aritmética 2.	Alterar entradas da memória.	Acrescentar um multiplicador à unidade funcional.	Alterar a unidade aritmética 1.	Transferências entre registos no RF.

1. Alterar entradas da memória:

Altere o data path de forma se possam utilizar como entradas de endereço da memória os resultados da unidade funcional. Mais concretamente acrescente dois MUXs na entrada desta controlados pelas entradas MMA e MMB, tal como representado na figura que se segue:

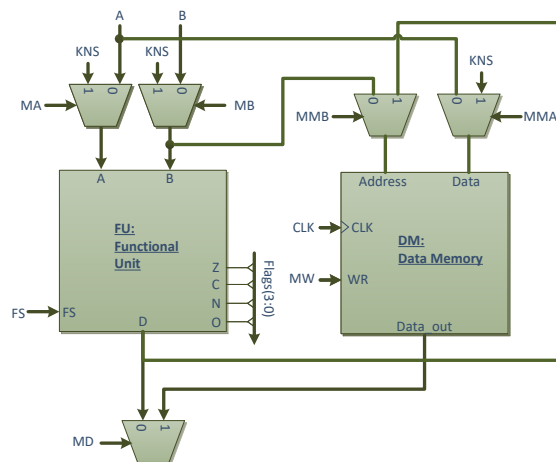


Figura 2 – Diagrama de blocos com alteração das entradas da memória.

a) Teste as alterações ao data path, nomeadamente implementando as seguintes operações:

```

R1 ← 38h
R2 ← 62h
M[4Ch + R1] ← R2
R1 ← R1 + 1
R3 ← M[4Bh + R1]
  
```

2. Alterar a unidade aritmética 1

Altere a unidade aritmética para reduzir o consumo para operações com números de valor absoluto reduzido colocando os bits de maior peso a zero, mesmo no caso em que os números são negativos. Adicione lógica que deteta que a parte alta do A e B (os 8 bits mais significativos) é igual ao bit de sinal, A(7) e B(7) ativando o sinal small_number. Se small_number estiver a um então coloque a parte alta (8 bits) de A e B a zero e faça extensão de sinal do resultado (8 bits). Tal está ilustrado no diagrama que se segue:

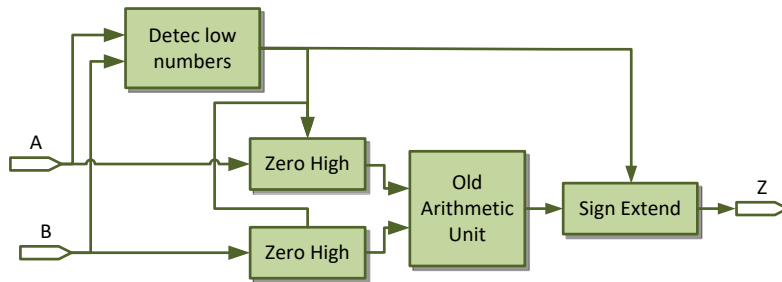


Figura 3 – Diagrama para as alterações da unidade aritmética 1.

Mais concretamente faça:

```

small_numberA = ( A(15) and A(14) ... A(7) ) or ( not A(15) and not A(14) ... not A(7) );
small_numberB = ( B(15) and B(14) ... B(7) ) or ( not B(15) and not B(14) ... not B(7) );
small_number = small_numberA and small_numberB;
  
```

Note que, como está indicado em cima, para os números pequenos deve os 8 bits mais significativos das entradas iguais a zero de modo a restarem 8 bits. Depois use 8 bits do resultado e faça extensão de sinal usando o bit Z(7) como sendo o bit de sinal.

a) Teste as alterações ao data path, nomeadamente implementando as seguintes operações:

```

R1 ← 56          (0038h)
R2 ← -60         (FFC4h)
R3 ← R1 + R2
R4 ← R1 - R2
  
```

Acrescente na simulação os sinais internos com os valores das entradas e das saídas da unidade aritmética antiga.

3. Transferências entre registos no RF.

Altere o register file (RF) de forma a que este implemente uma nova operação que permite copiar valores entre registos sem recurso à funcional unit (FU). Nomeadamente crie uma nova entrada do register file com designação FS e uma nova entrada no DataPath com designação FSRF que substituem as entradas WR. As funções executadas no register file em função de FS estão representadas na Tabela 4.

Tabela 4 – Funções do Register File alterado

FSRF	Função
00	Nenhuma operação
01	RD ← D
10	RD ← RA
11	RD ← RB

a) Teste as alterações ao data path, nomeadamente implementando as seguintes operações:

```

R1 ← 76  (4Ch)
R2 ← 56  (38h)
R3 ← R1  (através de RA)
R4 ← R2  (através de RB)
  
```

Acrescente à simulação as saídas dos registos R3 e R4 para que se possa observar o resultado das operações.

4. Acrescentar um multiplicador à unidade funcional.

Acrescente uma operação de multiplicação à unidade funcional. Para tal utilize o ficheiro multiplier.vhd fornecido na página da cadeira que recebe dois operandos de 16 bits com sinal e devolve um número de 32 bits com sinal com o resultado da multiplicação. No entanto note que deve guardar o resultado num registo pelo deve aproveitar apenas a

parte baixa do resultado e ativar a flag de overflow caso apropriado. Deve modificar a operação 1111b da unidade funcional e manter as restantes inalteradas.

a) Teste as alterações ao data path, nomeadamente implementando as seguintes operações:

```

R1 ← 76          (004Ch)
R2 ← 56          (0038h)
R3 ← -30000     (8AD0h)
R4 ← R1 * R2
R5 ← R2 * R3
  
```

Não se esqueça de verificar o resultado da flag de overflow.

5. Alterar a unidade aritmética 2.

Altere a unidade aritmética para que seja capaz de efetuar duas operações de 8 bits simultâneas em vez de uma única operação de 16 bits. Para tal acrescente um bit ao sinal de FS da unidade aritmética de forma a ficar com 8 em vez de 4 operações. Altere o MUX da unidade funcional para selecionar a unidade aritmética para todos estes casos removendo 4 funções da unidade lógica, mas mantendo todas as restantes funções inalteradas. Sempre que o bit mais significativo do FS da unidade aritmética estiver a 1 devem ser efetuadas duas operações de 8 bits enquanto que quando está a zero devem-se manter operações de 16 bits pré-existentes. A unidade aritmética deve ser escolhida sempre que FS for da forma 0xxx.

Para efetuar as operações de 8 bits devem interromper a cadeia de transporte existente nos somadores completos (full-adders), substituindo o carry in da parte alta do somador pelo valor anteriormente atribuído ao carry zero, tal como representado na Figura 4.

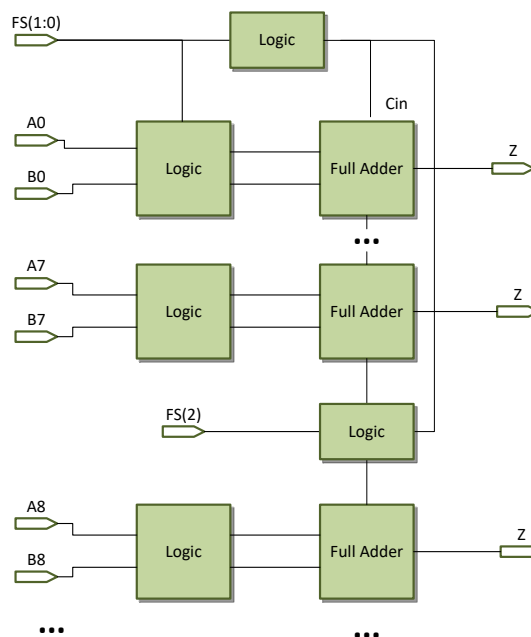


Figura 4 – Diagrama para as alterações da unidade aritmética 2.

a) Teste as alterações ao data path, nomeadamente implementando as seguintes operações:

```

R1 ← 08FFh
R2 ← 0101h
R3 ← R1 + R2 (8 bits)
R4 ← R1 + R2 (16 bits)
R5 ← R1 - R2 (8 bits)
R6 ← R1 - R2 (16 bits)
  
```

4 FASEAMENTO DO TRABALHO E RELATÓRIO

O trabalho decorrerá durante 2 semanas. As aulas de laboratório destinam-se, essencialmente, à correção de erros e à demonstração do trabalho efetuado. O trabalho deverá ser planeado e desenvolvido fora do horário de laboratório, sendo impossível realizá-lo sem preparação prévia apenas durante as 3 horas de laboratório. Tendo em consideração estes aspetos, o trabalho desenvolvido será avaliado por fases. Assim:

1ª SEMANA

- ❑ O objetivo para a 1ª aula de laboratório é o de caracterizar a Unidade de Processamento fornecida.

2ª SEMANA

- ❑ O objetivo para a 2ª aula de laboratório é o de alterar as funcionalidades da Unidade de Processamento fornecida.
- ❑ O relatório final será entregue até sexta-feira da segunda semana de laboratório, ou seja, **até dia 24 de Março**, via Fénix, em formato “pdf”. Este deverá ser sucinto e legível, em formato A4 e com tamanho de letra não inferior a 10 pontos. A sua **dimensão não deve exceder as 10 páginas** (excluindo a capa, com penalização de 1 valor por página adicional), mas deverá estar estruturado e conter a informação que se descreve de seguida:
 - Capa
 - Identificação do trabalho e autores.
 - Primeira parte
 - Análise e caracterização da UP, com apresentação de tabelas que caracterizem as unidades UF e UA, completamente preenchidas e devidamente justificadas.
 - Resposta a todas as questões indicadas na secção 2.
 - Segunda parte
 - Alterações efetuadas na Unidade de Processamento, nomeadamente os esquemas gerados pelo Vivado assim como o código VHDL.
 - Os resultados das simulações que confirmam o funcionamento do circuito.
 - Resposta a todas as questões indicadas na secção 3.
 - **O projeto do Vivado** com as alterações efetuadas, gerado pelo comando “File→Archive Project” do Vivado.
 - Discussão do trabalho
 - Comentários ao trabalho realizado e alterações efetuadas no laboratório.

Nota: NÃO serão valorizados esquemas ou simulações ilegíveis ou não comentados.

5 AVALIAÇÃO DO TRABALHO

A **avaliação** do trabalho será realizada, **ao longo das duas aulas de laboratório** de acordo com as cotações indicadas no enunciado. Valoriza-se, em particular: (1) a participação e empenho dos alunos nas aulas; (2) a originalidade da solução e dos testes realizados; (3) o grau de detalhe da solução; (4) a descrição do processo de síntese e justificação das diferentes opções de projeto; (6) a boa estruturação, a escrita sucinta e objetiva e, ainda, a boa apresentação do relatório.

6 BIBLIOGRAFIA

[1] N. Horta, “Arquitecturas de Computadores”, Aulas Teóricas, 2014.

[2] M. Morris Mano, Charles R. Kime, “Logic and Computer Design Fundamentals”, 2nd Edition Updated, Prentice-Hall International, 2008.

[3] G. Arroz, J. Monteiro, A. Oliveira, “Arquitectura de Computadores: dos Sistemas Digitais aos Microprocessadores”, IST Press, 2007.