



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Arquitetura e Organização de Computadores 1

Prática 4 Memória e Registradores

Professores: Luciano de Oliveira Neris e Mauricio Fernandes Figueiredo

Autores (Grupo R)

Guilherme Campos Marques, 727338 e Engenharia da Computação

Leticia Bossatto Marchezi, 791003 e Ciência da Computação

Marcos Cardoso Vendrame, 790725 e Ciência da Computação

Mateus Grota Nishimura Ferro, 771043 e Ciência da Computação

São Carlos, 17 de Abril de 2021

1. Experimentos com memória de dados

Utilizou-se do ambiente de simulação online [EDA Playground](#) para a implementação do testbench da memória de dados, memória esta que necessariamente precisa ser separada da memória de instruções, pelo motivo de as instruções ocorrerem em um único ciclo.

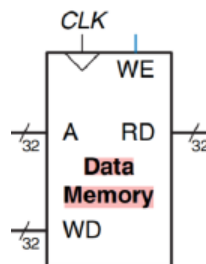


Imagem 1: Memória de dados.

Desta forma, foi definido dentro do testbench, além do clock, quatro variáveis do tipo logic, sendo elas:

WE(write enable): determinará se o conteúdo da entrada WD será escrito no endereço A.

WD(write data): receberá o conteúdo caso seja gravação.

RD: saída que disponibilizará o conteúdo do endereço A, caso seja leitura.

A: endereço a que se refere a operação(leitura ou escrita).

Tendo as variáveis definidas, foram testados alguns casos com o intuito de testar as operações de leitura e escrita na memória de dados.

Usando o primeiro caso para exemplificar:

```
we = 1; // write enable
wd = 531; // valor a ser salvo
a = 16; // endereco para escrita
```

Imagem 2: Escrita na memória de dados.

Como o sinal de controle WE é 1, haverá uma escrita na memória de dados, o conteúdo dessa escrita é definido por WD e o endereço no qual WD será escrito é especificado por A.

```
we = 0; // leitura de valor
a=34; // endereco lido
```

Imagem 3: Leitura na memória de dados.

Como o sinal de controle WE é 0, será realizada uma leitura no endereço especificado por A.

Em relação ao comportamento da memória para escrita e leitura em um mesmo tempo, foi verificado que nas operações de escrita já ocorre a leitura simultaneamente. Por exemplo, no tempo=100ns, ocorre a escrita do valor 0x36 no endereço 0x20; ao mesmo tempo, a leitura foi realizada e o sinal de rd foi atualizado para 0x36.

Assim, temos um testbench atestando o correto funcionamento da memória de dados através das operações de leitura e escrita em diferentes endereços, tendo como resultado a seguinte tabela:

time	write enable	endereco	entrada wd	saida rd	Operacao
20	1	00000010	00000213	00000213	Escrita
40	1	00000022	00000213	00000213	Escrita
60	0	00000022	00000213	00000213	Leitura
80	0	00000010	00000213	00000213	Leitura
100	1	00000020	00000036	00000036	Escrita
120	1	0000000c	00000028	00000028	Escrita
140	0	00000020	00000028	00000036	Leitura
160	0	0000000c	00000028	00000028	Leitura
180	1	0000001e	00000042	00000042	Escrita
180	1	0000001e	00000042	00000042	Leitura

Imagem 4: Tabela resultante.

2. Experimento com banco de registradores

Utilizou-se do ambiente de simulação online [EDA Playground](https://eda-playground.com/) para a implementação do testbench do banco de registradores, componente utilizado para armazenar o conteúdo dos 32 registradores definidos.

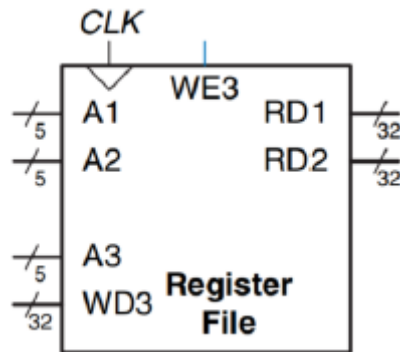


Imagem 5: Banco de registradores.

Desta forma, foi definido dentro do testbench, além do clock, quatro variáveis do tipo logic, sendo elas:

WE3(write enable): sinal de controle para habilitar ou desabilitar a escrita.

WD3(write data): valor que vai ser armazenado no A3.

A1: registrador de leitura.

A2: registrador de leitura.

A3: registrador em que será gravado o conteúdo.

RD1(read data): conteúdo do registrador A1.

RD2(read data): conteúdo do registrador A2.

Tendo as variáveis definidas, foram testados alguns casos com o intuito de testar as operações de leitura e escrita no banco de registradores.

Usando o primeiro caso para exemplificar:

```
wd3 = 225; // valor a ser armazenado
a3 = 8; // endereço
we3 = 1; // write enable
```

Imagem 6: Exemplo de escrita no banco de registradores.

Como o sinal de controle WE3 é 1, haverá uma escrita em um registrador, o endereço do registrador destino está armazenado em A3 e o conteúdo que será armazenado está em WD3.

```
we3 = 0; // load
a1 = 8; // endereço

we3 = 0;
a2 = 5;
i=i+1;
```

Imagem 7: Exemplo de leitura no banco de registradores.

Como o sinal de controle WE3 é 0, haverá a leitura de registradores, os dois registradores que são lidos estão armazenados no endereço 8 e 5 respectivamente.

Além disso, é possível verificar o comportamento do registrador de endereço 7, no ciclo 5, em que no tempo=80ns ocorre a escrita do valor 0x00012fd1. No mesmo instante não há leitura, somente após 10ms que seu valor será lido na borda de descida do *clock*.

Assim, temos um testbench atestando o correto funcionamento do banco de registradores através das operações de leitura e escrita em diferentes endereços, tendo como resultado a seguinte tabela:

time	a1	a2	a3	wd3	rd1	rd2	we3	Ciclo
0	xx	xx	08	000000e1	xxxxxxxx	xxxxxxxx	1	Escrita registrador 8 - ciclo 1
20	08	xx	08	000000e1	xxxxxxxx	xxxxxxxx	0	Leitura registrador 8 - ciclo 2
40	08	xx	05	000d7fad	000000e1	xxxxxxxx	1	Escrita registrador 5 - ciclo 3
60	08	05	05	000d7fad	000000e1	xxxxxxxx	0	Leitura registrador 5 - ciclo 4
80	08	05	07	00012fd1	000000e1	000d7fad	1	Escrita no reg. 7- ciclo 5
90	07	05	07	00012fd1	000000e1	000d7fad	0	Leitura do reg. 7- ciclo 5
120	07	05	02	00000011	00012fd1	000d7fad	0	Escrita no reg2 sem we3 - ciclo 6
140	02	05	02	00000011	00012fd1	000d7fad	0	Leitura do reg 2 - ciclo 7

Imagem 8: Tabela resultante.

3. Referências Bibliográficas:

1.Projeto Mem - <https://edaplayground.com/x/uL9N>

2.Projeto RegFile - <https://edaplayground.com/x/nwMW>

Tocci, Ronald J. - **Sistemas digitais : princípios e aplicações** / Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss ; revisão técnica Renato Giacomini ; tradução Jorge Ritter. – 11. ed. – São Paulo : Pearson Prentice Hall, 2011.