O datapath é constituido por 9 blocos , os quais vou passar a explicar a maneira como pensei para os modelar.

TABELA DE CONVERSAO:

A tabela de conversão demonstrada foi feita com base nos exemplos apresentados no guião de apoio ao projeto final.

PROGRAM COUNTER:

O PC, program counter, é o contador que guarda o endereço da próxima instrução a ser executada.

Começei por defenir as entradas e saídas do pc com base no guião do projeto , e depois com a ajuda dos guiões das aulas teóricas modelei o corpo do program counter.

ALU:

O bloco Alu tem como principal função realizar as operações da tabela que podem ver aqui.

Na primeira parte do codigo da alu estão definidas as entradas e saídas do bloco , bem como os seus vetores lógicos.

Depois disso estão definidos os sinais de cada entrada/saida.

No corpo da alu defini as operaçoes que aparecem na tabela , por ordem.

IMEMORY:

Tem como principal função guardar as instruçoes de um programa.

No código modelado em VHDL da Imemory, são apresentadas as entradas e saídas do bloco na primeira parte.

Depois disso, são apresentadas as instruçoes do tipoI e do tipo II pedidas no enunciado do projeto que também podem ser vistas na Tabela de conversao das instruçoes para código maquina.

Dmemory:

A função da Dmemory é guardar os dados de um programa.

Incicialmente comecei por defenir entradas e saídas , de seguida defini na arquitetura o tamanho/quantidade de palavras que o programa ia ler.Em seguida , criei um array para armazenar todas essas palavras com 8 bits cada uma.

Inicializei a memória com os valores pedidos no enunciado e em seguida no processo do clock fiz aquilo que pensei para WD.

REGISTER:

O registers é um bloco de 8 registos respetivamente de 8 bits cada.

Na entity são definidas as entradas e saídas do register.

Criei um array para guardar os registos efetuados (do 0 ao 7).

No processo do clock e reset é estudado o sinal do register.

No final do processo, os adress’s dos registos são passados para unsigned e para integer para serem lidos e armazenados.

Não consegui fazer a simulação da tb porque me ocorreram erros que não consegui resolver , mas fica aqui print da simulação do bloco.

SIGNEXTEND:

A função do SignExtend é extender um vetor que inicialmente é de 7 bits e transforma-lo num vetor de 8 bits.

A implementaçao do SignExtend foi uma das mais fáceis comparadas com alguns blocos do projeto pois foi bastante rápido e simples de realizar.

Comecei por definir a quantidade de bits de entrada e a quantidade de bits de saída.

Depois disso criei um processo com base no input para saber qual seria o output do meu SignExtend.

Não achei necessária a criação de uma TB para o SignExtend visto que foi uma implemetaçao tao básica e rápida.

Mux2:1:

A função do mux2\_1 é transformar duas entradas em uma so saída.

Achei também bastante fácil a modelação do mux pois já o fizemos varias vezes nas aulas e também nos slides pude ver como se modelava o mesmo , então também não achei oportuno a criação de uma tb.

FASE2:

Esta fase consiste na implementaçao e teste da unidade de controlo.

Criei a control unit e gerei um diagrama de estados da própria unidade de controlo(como pedido no enunciado).

Obtive também ajuda nos slide da TP da preparação para o projeto final.

No diagrama existem 6 estados e cada um deles foi instanciado na control unit.

O estado fetch le a instrução da memoria de instruçoes e atualiza o program counter.

Caso o EnPc seja 1 e Ri seja 1 , avançamos para o próximo estado.

No decode li o conteúdo dos registos como pedido , e o único caso em que o estado não avança é quando o OpCode é 000 , ou seja instrução NOP.

Avançando para o próximo estado, Execute, modelei 3 calculos da ALU dos quais 1 do tipo I e dois(ADDI E LW) do tipo dois.

No estado seguinte,WriteMem,guardei os resultados na memoria , atraves do calculo SW do tipo II ( como pedido no enunciado).

Apliquei o memso para o estado RegUpdate.

Fase3:

Nesta terceira fase foi pedido para interligar o processador completo , ou seja , ligar o datapath À control unit e executar o seu teste no simulador.

Comecei por instanciar primeiro toda a fase1(datapath) e so depois adicionar a control unit e liga-la consequentemente.

Criei signals para simular os fios , e interliguei-os com base na figura que nos foi fornecida no guião.

Criei também um “processo” para instanciar cada bloco e ligá-lo aos outros com os respetivos “fios”.