Instrução DIV

Nome: José Kilson de Souza Oliveira

Matricula: 0810847/1

Na arquitetura do PD-8 crie uma instrução em seu assembler:

1. Dê o Fluxo Geral da instrução na arquitetura do processador;
2. Dê o Fluxo Detalhado da instrução gerando os sinais e ciclos de clocks;
3. Projete a máquina de estados de controle da arquitetura utilizando Verilog.
4. Fluxo Geral

MAH <- M[PC]

PC <- PC + 1

MAL <- M[PC]

PC <- PC + 1

MD <- M[MA]

AC <- MD

M[SP] <- RFLAGS

Push

Acumulador

SP <- SP - 1

M[SP] <- AC

SP <- SP - 1

AC <- SHR(AC)

AC <- SHR(AC)

AC <- AC + MD

AC <- SHR(AC)

SP <- SP + 1

AC <- M[SP]

SP <- SP + 1

RFLAGS <- M[SP]

MD <- AC

MD <- AC

AC <- SHR(AC)

MD <- MDL

Pop

Acumulador

AC <- MDH

MD < 1

1

0

ERROR

AC < MD

1

AC <- ACC

0

AC <- AC - MD

M[SP] <- RFLAGS

SP <- SP - 1

M[SP] <- AC

SP <- SP - 1

Push

Acumulador

AC <- ACC

ACC <- MD

INC

MD<- 1

AC <- AC + MD

MD<- ACC

ACC<- AC

SP <- SP + 1

AC <- M[SP]

SP <- SP + 1

RFLAGS <- M[SP]

Pop

Acumulador

OBS.: DIV2, DIV3, DIV4 e V são apenas “labels”.

1. Fluxo Detalhado

**T4**

ENPCA

**T5**

ENPCA

MR

1

ENPCA

MR

(LMAH)

**T6**

2

**T7**

(INCPC)

**T8**

ENPCA

ENPCA

MR

(LMAL)

ENPCA

MR

**T9**

3

**T10**

**T11**

(INCPC)

ENMAA

**T12**

ENMAA

MR

4

ENMAA

MR

(LMD)

**T13**

ENDES

(LAC)

5

**T14**

ENSPA

ENFLD

ENSPA

ENFLD

MW

(DECSP)

ENSPA

ENDES

ENSPA

ENDES

MW

(DECSP)

(LRESET)

(LINT)

**T15**

**T16**

**T17**

6 - Push

**T18**

**T19**

**T20**

ENDES

(LMD)

ENDES

OP2

(LAC)

(LRESET)

(LINT)

ENDES

(LMD)

ENDES

OP2

(LAC)

(LRESET)

(LINT)

**T21**

7 - 1° SHL

**T22**

**T23**

7 - 2° SHL

**T24**

ENDES

(LMD)

ENDES

OP2

(LAC)

(LRESET)

(LINT)

ENDES

(LMD)

ENDES

OP2

(LAC)

(LRESET)

(LINT)

**T25**

7 - 3° SHL

**T26**

**T27**

7 - 4° SHL

**T28**

ENDES

SD0

(LAC)

(LRESET)

(LINT)

ENDES

SD0

(LAC)

(LRESET)

(LINT)

ENDES

SD0

(LAC)

(LRESET)

(LINT)

**T29**

8 - 1° SHR

**T30**

8 - 2° SHR

**T31**

8 - 3° SHR

ENDES

SD0

(LAC)

(LRESET)

(LINT)

ENDES

(LMD)

**T32**

8 - 4° SHR

9

**T33**

**T34**

(INCSP)

**T35**

ENSPA

ENSPA

MR

(LAC)

ENSPA

MR

**T35**

**T36**

10 - Pop

**T38**

ENSPA

MR

ENSPA

(INCSP)

**T39**

**T40**

ENSPA

MR

(LINTE)

(LCARRY)

(LZERO)

(LRESET)

(LINT)

ENDES

SD0

(LAC)

(LRESET)

(LINT)

ENDES

SD0

(LAC)

(LRESET)

(LINT)

10 – Pop (cont.)

**T41**

**T42**

11 - 1° SHR

**T43**

11 - 2° SHR

ENDES

SD0

(LAC)

(LRESET)

(LINT)

**T44**

11 - 3° SHR

ENDES

SD0

(LAC)

(LRESET)

(LINT)

11 - 4° SHR

**T45**

1

MD <1

0

ERROR

T0

AC < MD

1

0

ENDES

LAC

ENDES

OP2

OP1

(LAC)

SCZ

(LCARRY)

(LRESET)

(LINT)

**T46**

ENSPA

ENFLD

ENSPA

ENFLD

MW

(DECSP)

ENSPA

ENDES

ENSPA

ENDES

MW

(DECSP)

(LRESET)

(LINT)

**T47**

**T48**

**T49**

**T50**

12 - push

**T51**

**T52**

ENDES

LAC

13 – carrega AC

**T53**

14 – carrega ACC

ENDES

LACC

**T54**

ENCG

CON

(LMD)

**T55**

ENDES

OP2

(LAC)

SCZ

(LCARRY)

(LZERO)

(LRESET)

(LINT)

15 - INC

**T56**

ENDES

LMD

ENDES

LACC

(INCSP)

ENSPA

MR

ENSPA

MR

(LAC)

ENSPA

(INCSP)

**T57**

16 – carrega MD

**T58**

17 – carrega ACC

**T59**

**T60**

**T61**

**T62**

18 - Pop

**T63**

**T64**

ENSPA

MR

ENSPA

**T65**

ENSPA

MR

(LINTE)

(LCARRY)

(LZERO)

(LRESET)

(LINT)

18 - Pop (cont.)

**T66**

1. Máquina de Estados

**module div(T, clock);**

**reg [32:0] T;**

**input clock;**

**output SIGNAL;**

**always @(posedge clock or negedge clock) begin**

**if (clock = 0)**

**begin**

**ENPCA = 0;**

**MR = 1;**

**LMAH = 0;**

**LMAL = 0;**

**INCPC = 0;**

**ENMAA = 0;**

**LMD = 0;**

**DECSP = 0;**

**ENDES = 0;**

**LAC = 0;**

**LMD = 0;**

**LAAC = 0;**

**LINT = 0;**

**LINTE = 0;**

**LRESET = 0;**

**SD0 = 0;**

**OP1 = 0;**

**OP2 = 0;**

**ENSPA = 0;**

**ERROR = 0;**

**end**

**case (T) //begin**

**4: begin**

**ENPCA = 1;**

**T = 5;**

**end**

**5: begin**

**ENPCA = 1;**

**MR = 0;**

**T = 6;**

**end**

**6: begin**

**ENPCA = 1;**

**MR = 0;**

**LMAH = 1;**

**T = 7;**

**end**

**7: begin**

**INCPC = 1;**

**T = 8;**

**end**

**8: begin**

**ENPCA = 1;**

**T = 9;**

**end**

**9: begin**

**ENPCA = 1;**

**MR = 0;**

**T = 10;**

**end**

**10: begin**

**ENPCA = 1;**

**MR = 0;**

**LMAL = 1;**

**T = 11;**

**end**

**11: begin**

**INCPC = 1;**

**ENMAA = 1;**

**T = 12;**

**end**

**12: begin**

**ENMAA = 1;**

**MR = 0;**

**T = 13;**

**end**

**13: begin**

**ENMAA = 1;**

**MR = 0;**

**LMD = 1;**

**T = 14;**

**end**

**14: begin**

**ENDES = 1;**

**LAC = 1;**

**T = 15;**

**end**

**15: begin**

**ENSPA = 1;**

**ENFLD = 1;**

**T = 16;**

**end**

**16: begin**

**ENSPA = 1;**

**ENFLD = 1;**

**MW = 0;**

**T = 17;**

**end**

**17: begin**

**DECSP = 1;**

**T = 18;**

**end**

**18: begin**

**ENSPA = 1;**

**ENDES = 1;**

**T = 19;**

**end**

**19: begin**

**ENSPA = 1;**

**ENDES = 1;**

**MW = 0;**

**T = 20;**

**end**

**20: begin**

**DECSP = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 21;**

**end**

**21: begin**

**ENDES = 1;**

**LMD = 1;**

**T = 22;**

**end**

**22: begin**

**ENDES = 1;**

**OP2 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 23;**

**end**

**23: begin**

**ENDES = 1;**

**LMD = 1;**

**T = 24;**

**end**

**24: begin**

**ENDES = 1;**

**OP2 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 25;**

**end**

**25: begin**

**ENDES = 1;**

**LMD = 1;**

**T = 26;**

**end**

**26: begin**

**ENDES = 1;**

**OP2 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 27;**

**end**

**27: begin**

**ENDES = 1;**

**LMD = 1;**

**T = 28;**

**end**

**28: begin**

**ENDES = 1;**

**OP2 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 29;**

**end**

**29: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 30;**

**end**

**30: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 31;**

**end**

**31: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 32;**

**end**

**32: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 33;**

**end**

**33: begin**

**ENDES = 1;**

**LMD = 1;**

**T = 34;**

**end**

**34: begin**

**INCSP = 1;**

**T = 35;**

**end**

**35: begin**

**ENSPA = 1;**

**T = 36;**

**end**

**36: begin**

**ENSPA = 1;**

**MR = 0;**

**T = 37;**

**end**

**37: begin**

**ENSPA = 1;**

**MR = 0;**

**LAC = 1;**

**T = 38;**

**end**

**38: begin**

**INCSP = 1;**

**T = 39;**

**end**

**39: begin**

**ENSPA = 1;**

**T = 40;**

**end**

**40: begin**

**ENSPA = 1;**

**MR = 0;**

**T = 41;**

**end**

**41: begin**

**ENSPA = 1;**

**MR = 0;**

**LINTE = 1;**

**LCARRY = 1;**

**LZERO = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 42;**

**end**

**42: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 43;**

**end**

**43: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 44;**

**end**

**44: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 45;**

**end**

**45: begin**

**ENDES = 1;**

**SD0 = 1;**

**LAC = 1;**

**LRESET = 1;**

**LINT = 1;**

**if(MD < 1) begin**

**ERROR=1;**

**T = 0;**

**end**

**else begin**

**T = 46;**

**end**

**end**

**46: begin**

**if(AC>MD) begin // Instrução que faz a operação: AC = AC - MD**

**ENDES = 1;**

**OP2 = 1;**

**OP1 = 1;**

**LAC = 1;**

**SCZ = 1;**

**LCARRY = 1;**

**LRESET = 1;**

**LINT = 1;**

**T = 47;**

**end // fim da instrução**

**else begin // Instrução que carrega um dado no AC**

**ENDES = 1;**

**LAC = 1;**

**T = 0;**

**end // fim da instrução**

**end**

**default: begin**

**T=0;**

**end**

**47: begin // Instrução que insere o AC na pilha (PUSH)**

**ENSPA = 1;**

**ENFLD = 1;**

**T=48;**

**end**

**48: begin**

**ENSPA = 1;**

**ENFLD = 1;**

**MW =0;**

**T=49;**

**end**

**49: begin**

**DECSP=1;**

**T=50;**

**end**

**50: begin**

**ENSPA = 1;**

**ENDES = 1;**

**T=51;**

**end**

**51: begin**

**ENSPA = 1;**

**ENFLD = 1;**

**MW =0;**

**T=52;**

**end**

**52: begin**

**DECSP = 1;**

**LRESET = 1;**

**LINT =1;**

**T=53;**

**end // fim da instrução PUSH**

**53: begin // Instrução que carrega um dado no AC**

**ENDES = 1;**

**LAC = 1;**

**T=54;**

**end // fim da instrução**

**54: begin // Instrução que carrega um dado no ACC**

**ENDES = 1;**

**LACC = 1;**

**T=55;**

**end // fim da instrução**

**55: begin // Instrução que Incrementa o AC**

**ENCG = 1;**

**CON = 1;**

**LMD = 1;**

**T=74;**

**end**

**56: begin**

**ENDES = 1;**

**OP2 = 1;**

**LAC = 1;**

**SCZ = 1;**

**LCARRY = 1;**

**LZERO = 1;**

**LRESET = 1;**

**LINT = 1;**

**T=57;**

**end // fim da instrução**

**57: begin // Instrução que carrega um dado no MD**

**ENDES = 1;**

**LMD = 1;**

**T=58;**

**end // fim da instrução**

**58: begin // Instrução que carrega um dado no ACC**

**ENDES = 1;**

**LACC = 1;**

**T=59;**

**end // fim da instrução**

**59: begin // Instrução que retira o AC da pilha (POP)**

**INCSP = 1;**

**T=60;**

**end**

**60: begin**

**ENSPA = 1;**

**T=61;**

**end**

**61: begin**

**ENSPA = 1;**

**MR = 0;**

**T=62;**

**end**

**62: begin**

**ENSPA = 1;**

**MR = 0;**

**LAC = 1;**

**T=63;**

**end**

**63: begin**

**INCSP = 1;**

**T=64;**

**end**

**64: begin**

**ENSPA = 1;**

**T=65;**

**end**

**65: begin**

**ENSPA = 1;**

**MR = 0;**

**T=66;**

**end**

**66: begin**

**ENSPA = 1;**

**MR = 0;**

**LINTE = 1;**

**LCARRY = 1;**

**LZERO = 1;**

**LRESET = 1;**

**LINT = 1;**

**T=46;**

**end // fim da instrução POP**

**default: begin**

**T=0;**

**end**

**endcase**

**//end**

**endmodule**