مریم علایی فرد 9631789`
علی ربیعی 9532267
کلیه دیتای مربوط به پردازنده در یک instance از کلاس Mips نگهداری می شود. کلاس Mips شامل یک RegisterFile است که رجیسترهای اصلی cpu را نگهداری میکند. همچنین شامل یک instance از کلاس نگهدارنده ی دیتای تمام stage ها هست. در هر مرحله روی instance هر stage تابع run عملیات مورد نیاز آن stage را انجام می دهد و تابع tick روی رجیسترهای مربوط به هر stage صدا زده می شود و در صورت نیاز مقادیر جدید را وارد رجیسترهای آن stage می کند.

```
:Decode
:bit of op code R type 6
:bit of fuct 6
OP_CODE  for alu=0

ADD_FUNCT = 0x20

SUB_FUNCT = 0x22

AND_FUNCT = 0x24

OR_FUNCT = 0x25

NOR_FUNCT = 0x27

SLT_FUNCT = 0x2a

:bit op code I type 6


;ADDI = 0x8

;ANDI = 0xC

;ORI = 0xD

;SLTI = 0xA
;BEQ = 0x4

;BNE = 0x5

;LW = 0x23

;SW = 0x2B

:bit op code  J type 6
```

;J = 0x2

JR_FUNCT = 0x8

;HLT = 0x3F

Pipeline Register use in this stage is: If_Id & Id_EX & RegFile & programCounter and it has method for stallPipeline and takeBranch & takeJump

It has mask method: the mask and the 32 bit with 11111100000000000000000000000000 and give you the opcode and with the shift right you can have the only 6 bit of opcode

It read data 1 and 2 from IF_ID reg and set it to ID_EX ReG
If we want valu from EX_mem reg and set it to ID_EX we should have

:And We set signal control for each instruction
:First all of them
REG_DST, 0

ALU_SRC, 0

MEM_TO_REG, 0

REG_WRITE, 0

MEM_READ, 0

MEM_WRITE, 0

BRANCH, 0

BRANCH_NE, 0

JUMP, 0

JUMP_SRC, 0

ALU_OP, 0

OP_CODE, NOP

مثال:

این برنامه یک file به عنوان ورودی می خواند

```
10001100000000010000000000010000
10001100000000010000000000010100
10001100000000011000000000011000
00001000000000000000000000000111
00000000000000000000000000001001
00000000000000000000000000001110
00000000000000000000000000001011
00000000001000100010000000100000
00000000100000110010100000100010
11111100000000000000000000000000
```

و این دستورات mips را handel می کند:

```
0 load mem[4] -> r1
1 load mem[5] -> r2
2 load mem[6] -> r3
3 jump -> 7
4 9
5 14
6 11
7 r4 = r1 + r2
8 r5 = r4 - r3
9 Halt
```

که 13 clockcycleطول می کشد این برنامه در خط 7و 8 هازارد داریم که با انجام fowrwarding هازارد را برطرف می کند و در clock cycle مشکلی بیش نمی اورد

و این ورودی:

```
10001100000000010000000000010000
10001100000000010000000000010100
10001100000000011000000000011000
00001000000000000000000000000111
00000000000000000000000000001001
00000000000000000000000000001110
00000000000000000000000000001011
00000000001000100010000000100000
00000000100000110010100000100010
10001100000010110000000000010100
00000000100010110010100000100010
11111100000000000000000000000000
```

شامل این دستور mips هست:

```
0 load mem[4] -> r1
1 load mem[5] -> r2
2 load mem[6] -> r3
3 jump -> 7
4 9
5 14
```

```
6 11
7 r4 = r1 + r2
8 r5 = r4 - r3
9 load mem[5] -> r11
10 r5 = r4 - r11
11 Halt
```

که شامل 16 clockcycle هست که در خط 9 و 10 هازارد داریم که با انجام forwardingو stallمشکل بر طرف می کند