

به نام خدا  
 فاز اول پروژه معماری کامپیوتر  
 سهیل نظری مندجین-هیربد بهنام-هیراد داوری-بنیامین ملکی  
 تیم ما برای انجام این پروژه تصمیم گرفت که پردازنده مبتنی بر معماری MIPS طراحی کند.  
 توضیحات فاز اول این پروژه به شرح زیر است.

## MIPS Instructions:

### R-Type:

لیست دستورات:

#	Instruction Name	Meaning	Func
1	XOR	$rd \leftarrow rs \wedge rt$	100110
2	SLL(Shift left logical)	$rd \leftarrow rt \ll \text{Sh.AMOUNT}$	000000
3	SLLV(shift left logical variable)	$rd \leftarrow rt \ll rs$	000100
4	SRL (Shift right logical) Unsigned right shift	$rd \leftarrow rt \gg \text{Sh.AMOUNT}$	000010
5	SUB	$rd \leftarrow rs - rt$	100010
6	SRLV(shift right logical variable)	$rd \leftarrow rt \gg rs$	000110
7	SLT	$rd \leftarrow rs < rt$ signed comparison	101010
8	Syscall	Finish cpu operation	001100
9	SUBU(Subtract unsigned)	$rd \leftarrow rs - rt$	100011
10	OR	$rd \leftarrow rs \vee rt$	100101
11	NOR	$rd \leftarrow rs \sim \vee rt$	100111
12	ADDu(Add unsigned)	$rd \leftarrow rt + rs$	100001
13	MULT	$rd \leftarrow rs \times rt$	011000
14	DIV	$rd \leftarrow rs \div rt$	011010
15	AND	$rd \leftarrow rs \& rt$	100100
16	ADD	$rd \leftarrow rs + rt$	100000
17	JR(Jump Reg)	$PC \leftarrow rs$	001000
18	SRA(signed right shift)	$rd \leftarrow rt \gg \text{Sh.AMOUNT}$	000011

### I-Type:

لیست دستورات:

#	Instruction Name	Meaning	Opcode
1	ADDi	$rt \leftarrow rs + \text{SIGN EXTEND (Imm)}$	001000
2	ADDiu(unsigned)	$rt \leftarrow rs + \text{SIGN EXTEND (Imm)}$	001001
3	ANDi	$rt \leftarrow rs \& \text{SIGN EXTEND (Imm)}$	001100
4	XORi	$rt \leftarrow rs \sim \text{SIGN EXTEND (Imm)}$	001110
5	ORi	$rt \leftarrow rs \vee \text{SIGN EXTEND (Imm)}$	001101
6	BEQ	$rs == rt: PC \leftarrow PC + \text{SIGN EXTEND(Imm} \mid \text{"00"})$	000100
7	BNE	$rs \neq rt: PC \leftarrow PC + \text{SIGN EXTEND (Imm} \mid \text{"00"})$	000101
8	BLEZ	$rs \leq 0: PC \leftarrow PC + \text{SIGN EXTEND (Imm} \mid \text{"00"})$	000110
9	BGTZ	$rs > 0: PC \leftarrow PC + \text{SIGN EXTEND (Imm} \mid \text{"00"})$	000111
10	BGEZ	$rs \geq 0: PC \leftarrow PC + \text{SIGN EXTEND (Imm} \mid \text{"00"})$	000001
11	LW	$rt \leftarrow \text{MEM} [rs + \text{SIGN EXTEND (Imm)}]$	100011
12	SW	$\text{MEM} [rs + \text{SIGN EXTEND (Imm)}] \leftarrow rt$	101011
13	LB	$rt[7:0] \leftarrow \text{MEM} [rs + \text{SIGN EXTEND (Imm)}]$	100000
14	SB	$\text{MEM} [rs + \text{SIGN EXTEND (Imm)}] \leftarrow rt[7:0]$	101000
15	SLTi	Set to 1 if Less, $rs < \text{SIGN EXTEND (Imm)}$ , $rt=1$	001010
16	Lui(load upper immediate)	The immediate value is shifted left 16 bits and store in register. The lower 16 bits are zeroes $rt \leftarrow \{\text{SIGN EXTEND (Imm)}, 0*16\}$	001111

## J-Type:

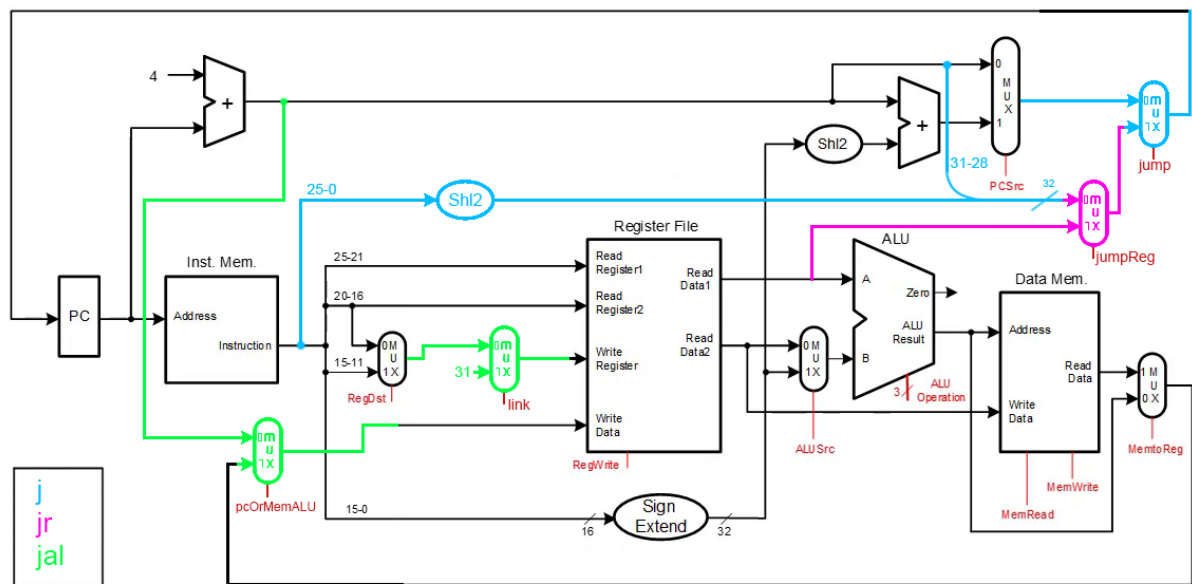
لیست دستورات:

#	Instruction Name	Meaning	opcode	Comments
۱	j	$PC \leftarrow \{(PC), \text{address}, 00\}$	000010	Jump to target address
2	<sup>h</sup> JAL	$R[31] \leftarrow PC$ then go to procedure address $PC \leftarrow \{(PC), \text{address}, 00\}$	000011	Use when making procedure call. This saves the return address in \$31

## CPU:

### Datapath:

مسیر داده طراحی شده برای پروژه با توجه به دستور های داده شده به شکل زیر است:



به نام خدا  
فاز اول پروژه معماری کامپیوتر  
سهیل نظری مندجین-هیربد بهنام-هیراد داوری-بنیامین ملکی

Controller:

سیگنال های مسیر داده توسط کنترلر با توجه به opcode دستور فعلی، func در حالت r-type، و خروجی zero واحد ALU تعیین می شوند. نمونه ایی از این سیگنال های کنترلی در cu

```
// J type opts
`J_TYPE:
  case (opcode)
    `J: begin
      Jump = 1'b1;
    end
    `JAL: begin
      {RegWrite, Jump} = 2'b11;
    end
  default: begin
    end
  endcase

casez (opcode)
// R type opts
`R_TYPE: begin
  case (func)
    `XOR: begin
      {RegDest, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_XOR;
    end
    `SLL: begin
      {RegDest, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_UNSIGNED_SHIFT_LEFT_SH_AMOUNT;
    end
    `SLLV: begin
      {RegDest, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_UNSIGNED_SHIFT_LEFT;
    end
    `SRL: begin
      {RegDest, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_UNSIGNED_SHIFT_RIGHT_SH_AMOUNT;
    end
```

```
// I type opts
default:
  case (opcode)
    `ADDi: begin
      {ALUsrc, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_ADD;
      SignExtend = 1;
    end
    `ADDiu: begin // similar to 'ADDi' in control sig
      {ALUsrc, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_ADD;
    end
    `ANDi: begin // DO NOT EXTEND SIGN
      {ALUsrc, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_AND;
    end
    `XORi: begin // DO NOT EXTEND SIGN
      {ALUsrc, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_XOR;
    end
    `ORI: begin // DO NOT EXTEND SIGN
      {ALUsrc, Link, RegWrite} = 3'b111;
      {ALUOp} = `ALU_OR;
```

به نام خدا  
فاز اول پروژه معماری کامپیوتر  
سهیل نظری مندرجین-هیربد بهنام-هیراد داوری-بنیامین ملکی

## Test Program:

پروژه ما در نهایت تمامی تست های انتخاب شده را نیز پاس می کند.

```
diff -u test/default/addtest.reg output/regdump.reg 1>&2  
make[1]: Leaving directory '/home/hirad/arch-proj/project-kiavash'  
[1;32mAll tests passed! (7 tests)[0m
```

برای مشاهده لاگ پاس شدن تست ها و محتوای رجیستر ها به فایل `verify-all` ضمیمه شده در گیت مراجعه کنید.