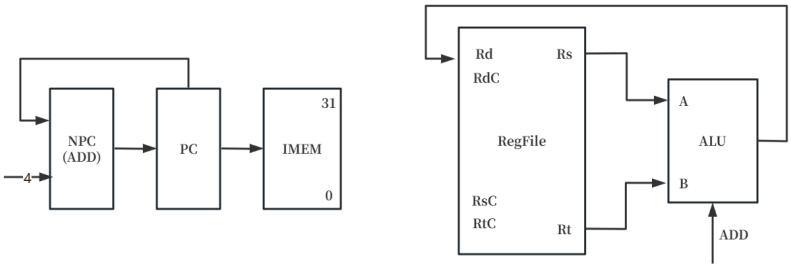
31条指令CPU

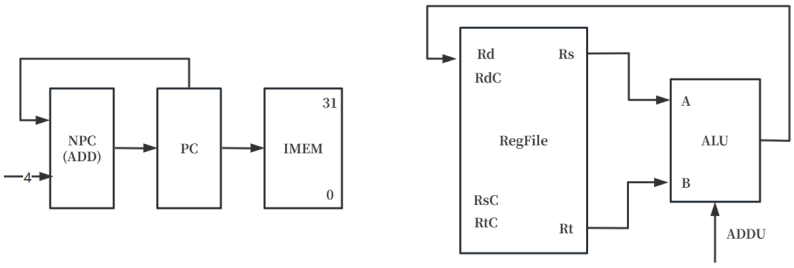
R型指令

1.ADD



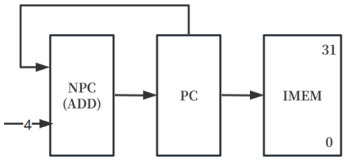

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A + B → RES)
RES → Rd
```

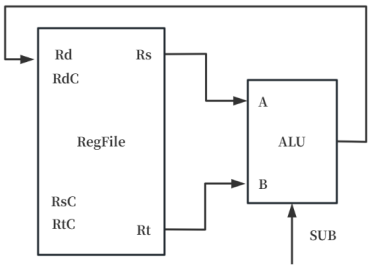
2.ADDU




```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A + B → RES)
RES → Rd
```

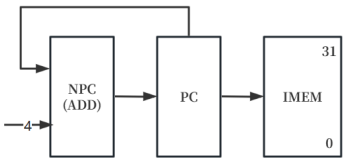
3.SUB

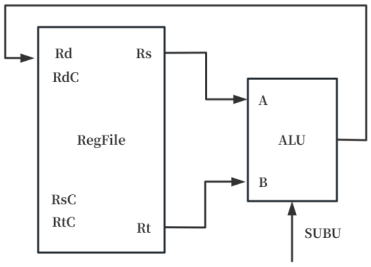





```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A - B → RES)
RES → Rd
```

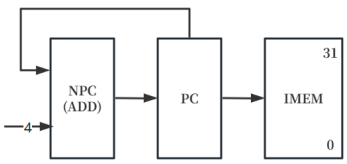
4.SUBU

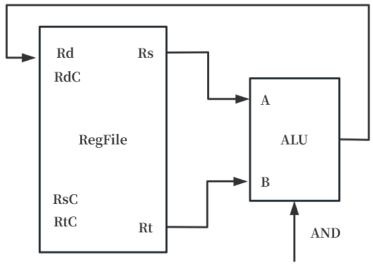





```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A - B → RES)
RES → Rd
```

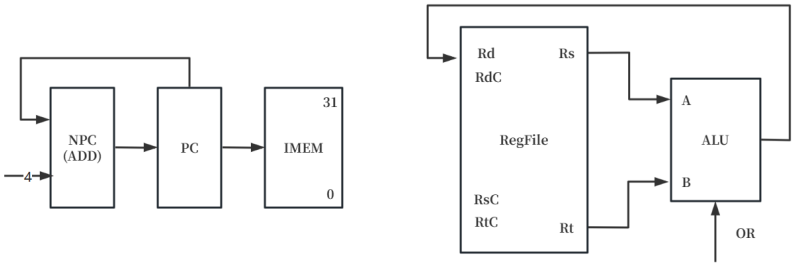
5.AND





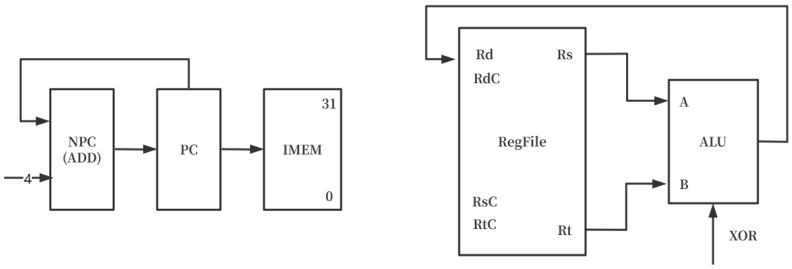
```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A & B → RES)
RES → Rd
```

6.OR



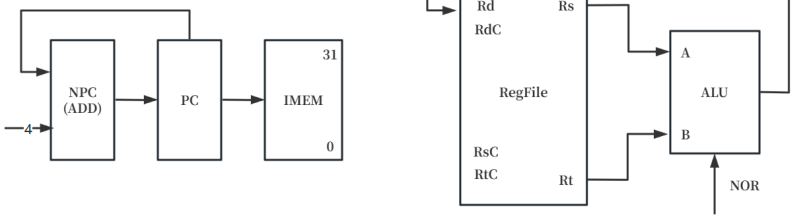

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A | B → RES)
RES → Rd
```

7.XOR



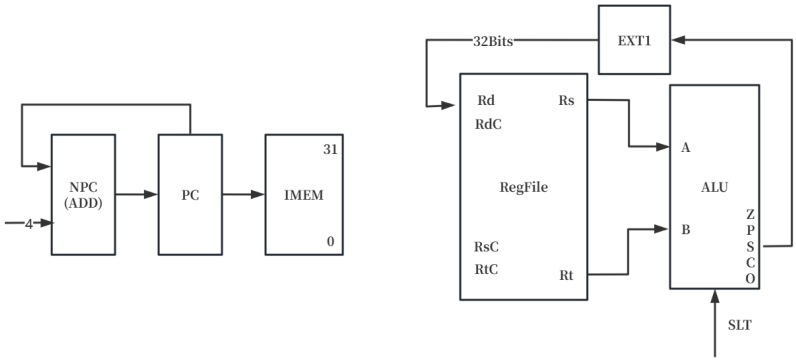

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A ⊕ B → RES)
RES → Rd
```

8.NOR



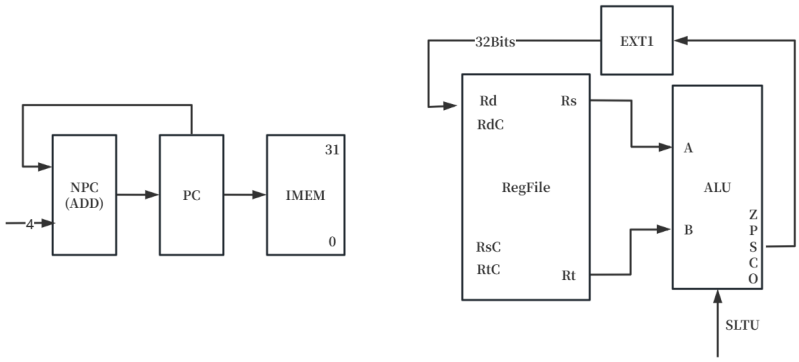

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A ⊙ B → RES)
RES → Rd
```

9.SLT



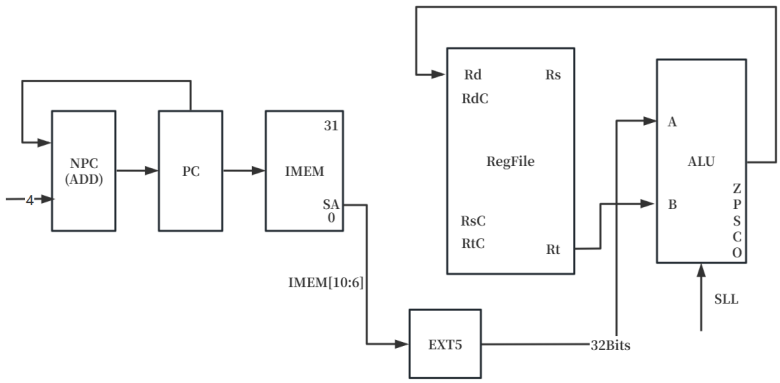

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A - B → RES)  //相减判断，负数则为Rs中数小
SF → EXT1      //注意要做扩展
EXT1_OUT → Rd
```

10.SLTU



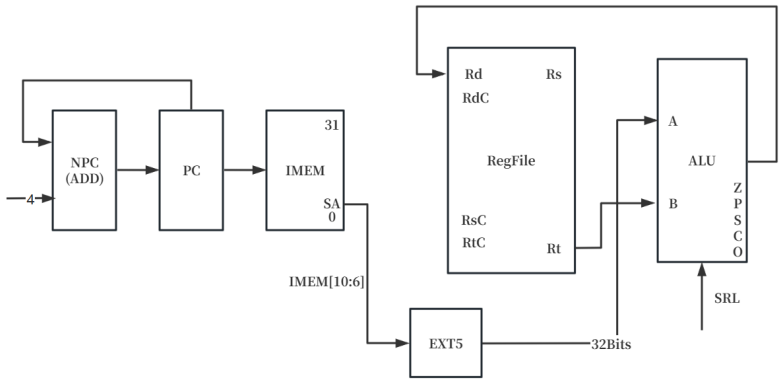

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → A , Rt → B
(A - B → RES)  //相减判断，负数则为Rs中数小
SF → EXT1      //注意要做扩展
EXT1_OUT → Rd
```

11.SLL



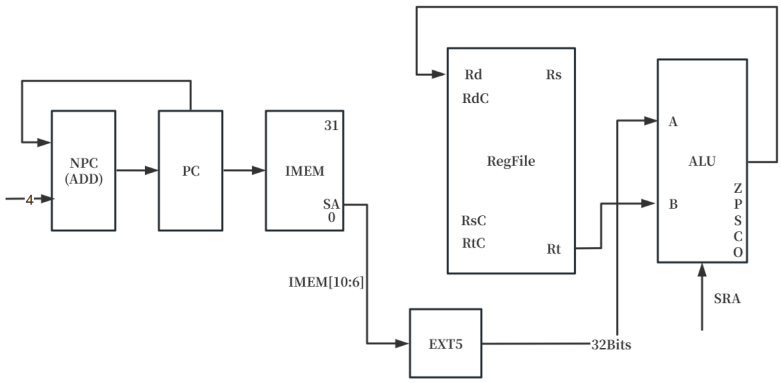

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[10:6] → EXT5
EXT5_OUT → A

Rt → B
(B<<A → RES)
RES → Rd
```

12.SRL



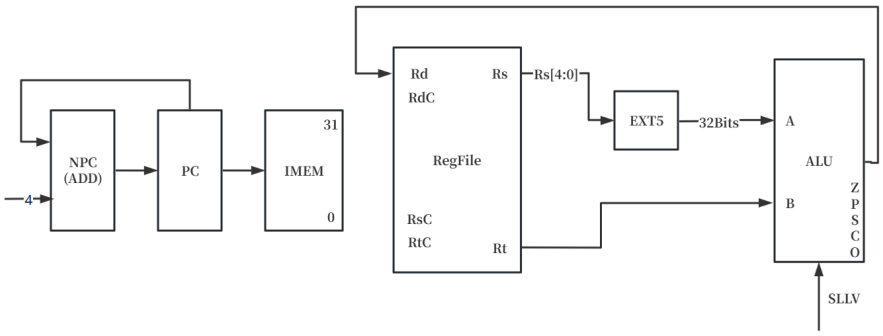

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[10:6] → EXT5
EXT5_OUT → A

Rt → B
(B>>A → RES)
RES → Rd
```

13.SRA




```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[10:6] → EXT5
EXT5_OUT → A

Rt → B
(B>>A → RES)
RES → Rd
```

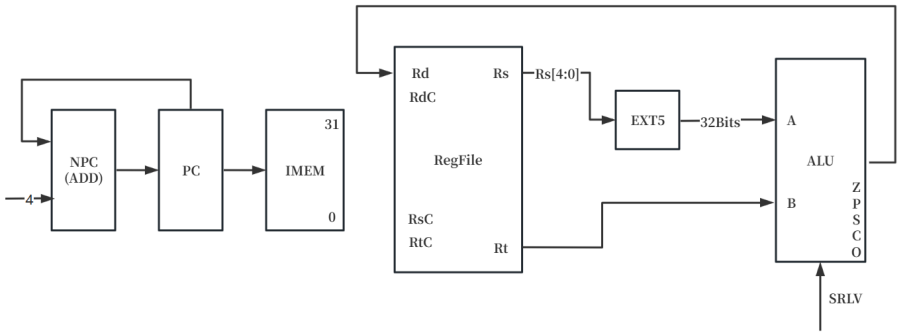
14.SLLV




```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs[4:0] → EXT5
EXT5_OUT → A
Rt → B
(B<<A → RES)
RES → Rd
```

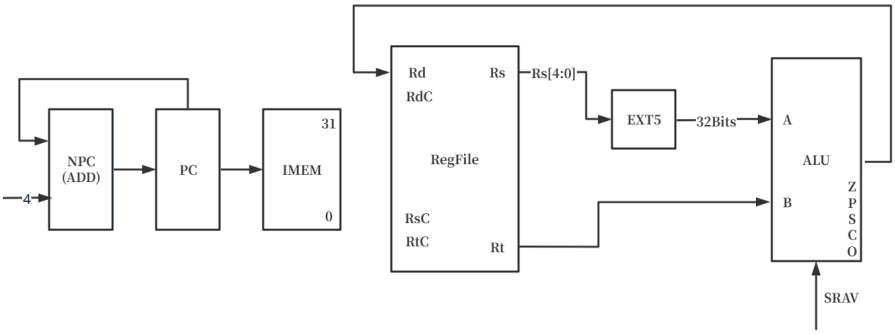
31

15.SRLV



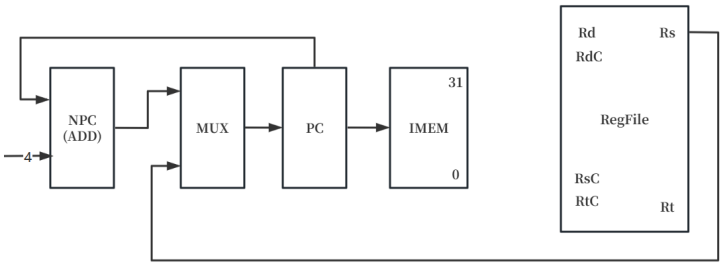

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs[4:0] → EXT5
EXT5_OUT → A
Rt → B
(B>>A → RES)
RES → Rd
```

16.SRAV




```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs[4:0] → EXT5
EXT5_OUT → A
Rt → B
(B>>A → RES)
RES → Rd
```

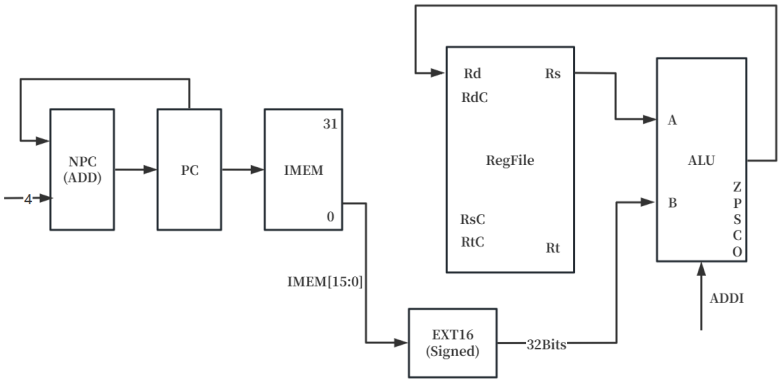
17.JR




```
PC → IMEM
PC + 4 → NPC    //无关指令
Rs → MUX
MUX_OUT → PC

NPC → MUX       //无关指令
```

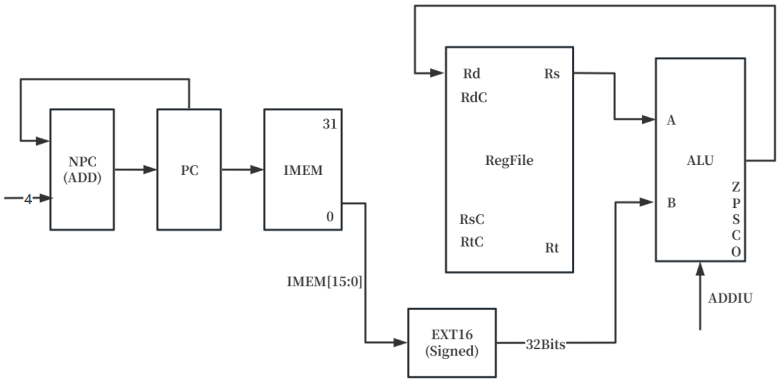
I型指令

18.ADDI



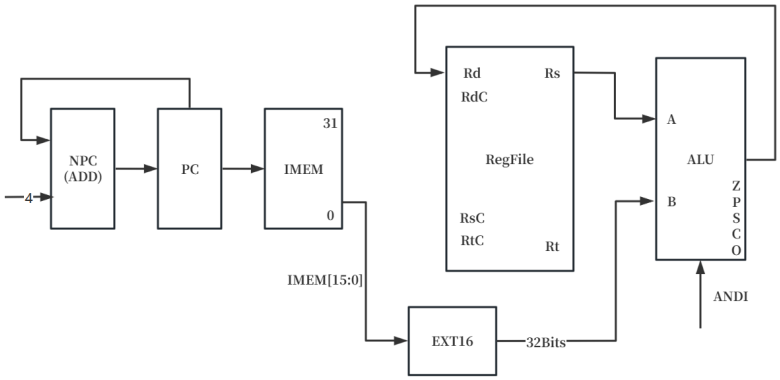

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A + B → RES)
RES → Rd
```

19.ADDIU



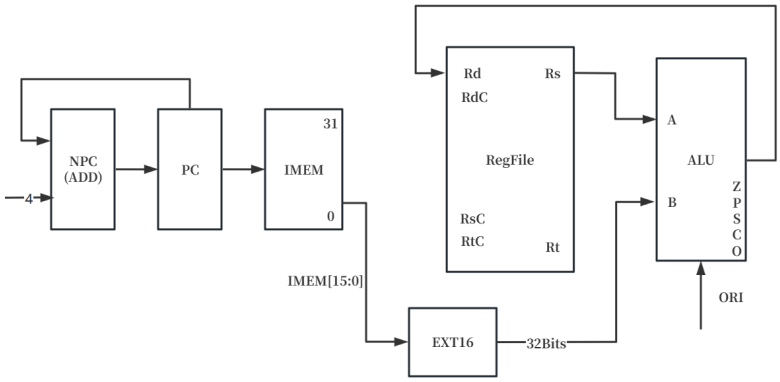

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A + B → RES)
RES → Rd
```

20.ANDI



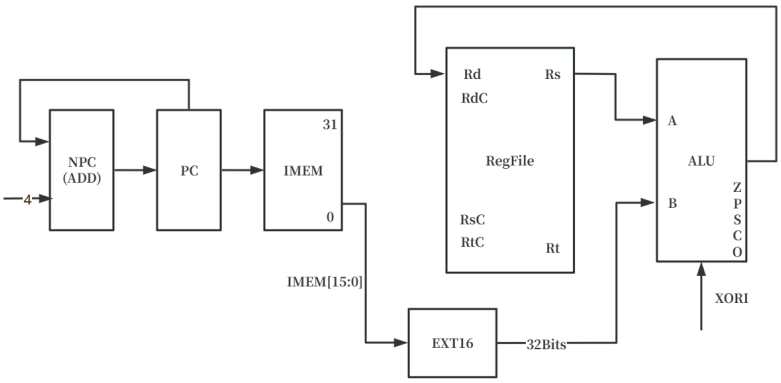

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A & B → RES)
RES → Rd
```

21.ORI



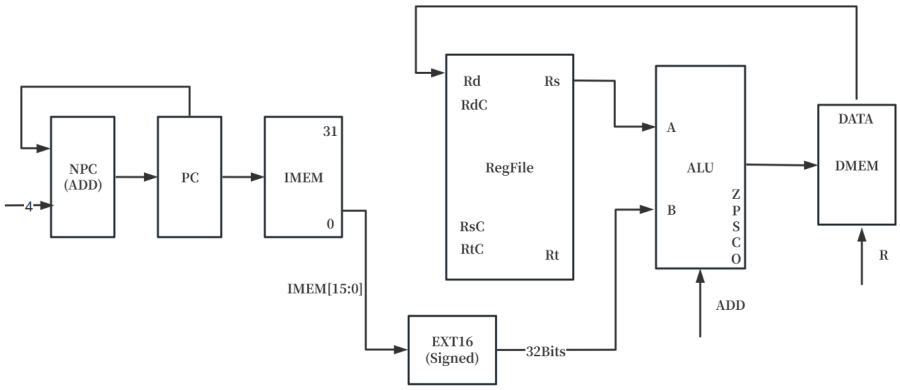

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A | B → RES)
RES → Rd
```

22.XORI



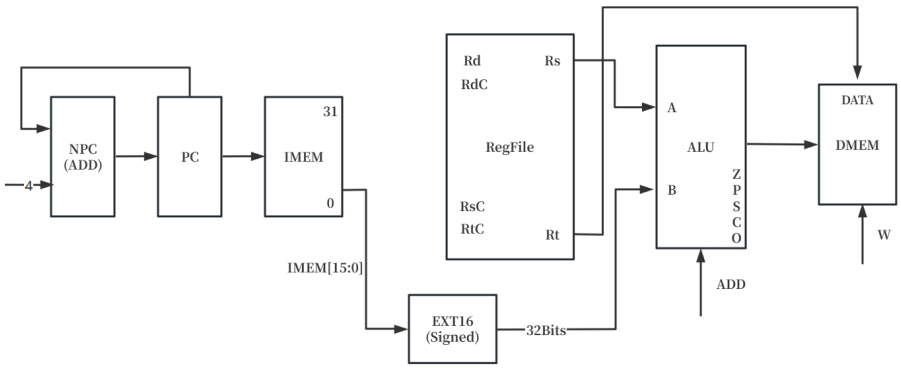

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A ⊕ B → RES)
RES → Rd
```

23.LW



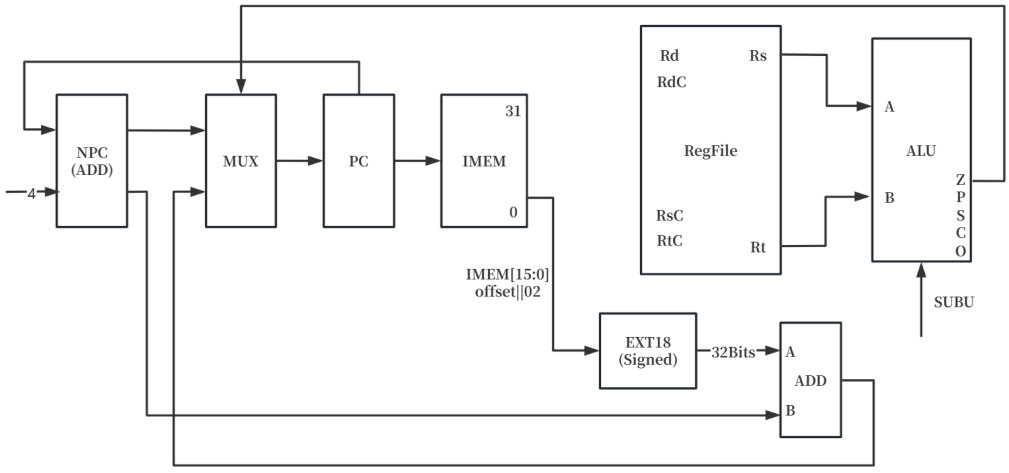

```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A + B → RES)
RES → DMEM_ADDR
DMEM_OUT → Rd
```

24.SW




```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A + B → RES)
Rt → DMEM
RES → DMEM_ADDR
```

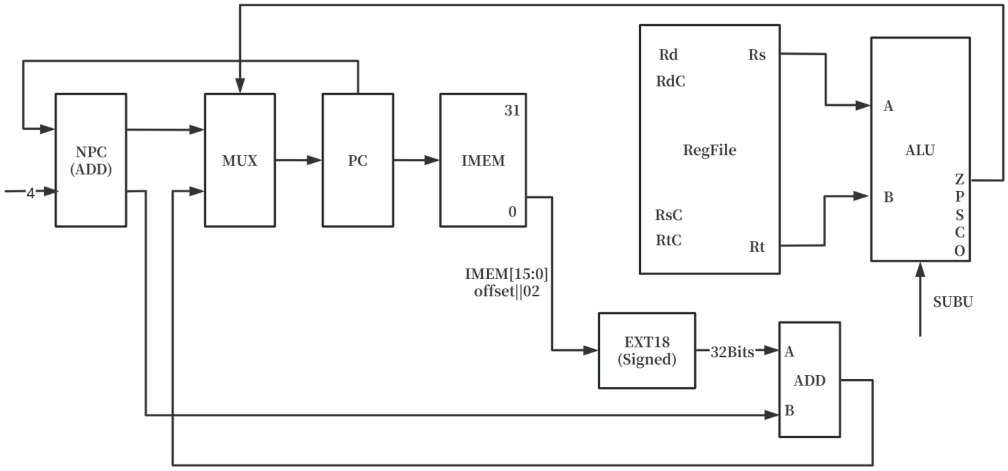
25.BEQ




```
PC → IMEM
PC + 4 → NPC
NPC → MUX

IMEM[15:0] || 02 → EXT18
EXT18_OUT → ADD_A
NPC → ADD_B
(ADD_A + ADD_B → ADD_OUT)
ADD_OUT → MUX

Rs → A
Rt → B
(A + B → RES)
```

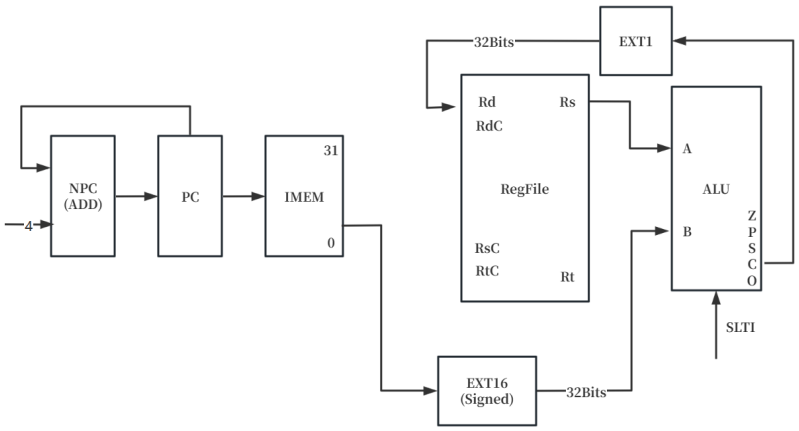
```
Z → MUX

MUX → PC
```

26.BNE



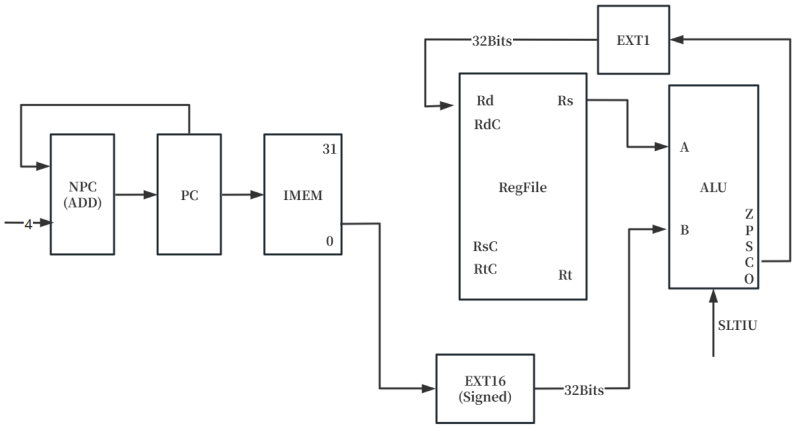

```
PC → IMEM
PC + 4 → NPC
NPC → MUX

IMEM[15:0] || 02 → EXT18
EXT18_OUT → ADD_A
NPC → ADD_B
(ADD_A + ADD_B → ADD_OUT)
ADD_OUT → MUX

Rs → A
Rt → B
(A + B → RES)
Z → MUX

MUX → PC
```

27.SLTI



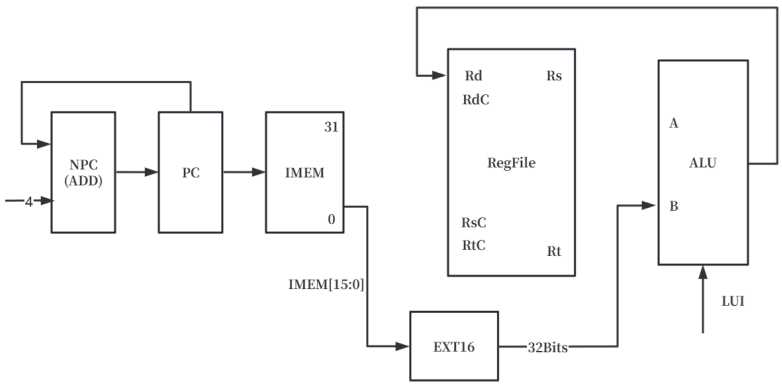
```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A - B → RES)
CF → EXT1
EXT1_OUT → Rd
```

28.SLTIU




```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
Rs → A
(A - B → RES)
CF → EXT1
EXT1_OUT → Rd
```

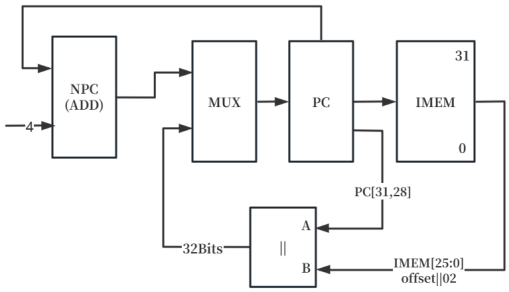
29.LUI



```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16
EXT16_OUT → B
RES → Rd
```

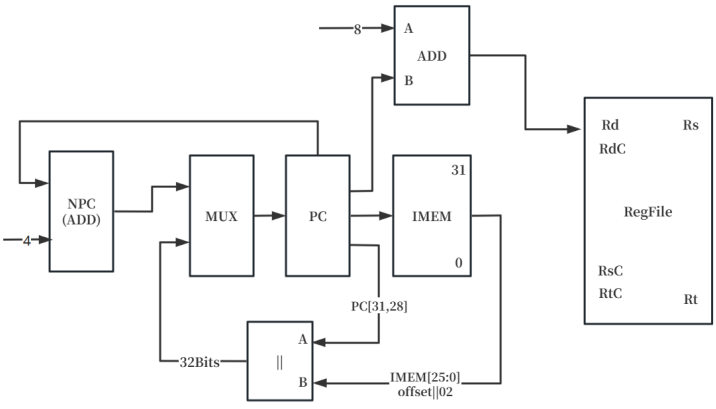
J型指令

30.J




```
PC → IMEM
PC + 4 → NPC
NPC → MUX

PC[31:28] → ||_A
IMEM[25,0] || 02 → ||_B
||_OUT → MUX

MUX_OUT → PC
```

31.JAL

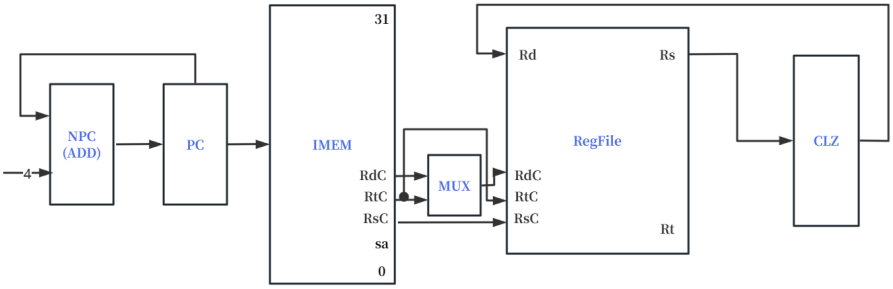



```
PC → IMEM
PC + 4 → NPC
NPC → MUX
```

```
8 → ADD_A
PC → ADD_B
(ADD_A + ADD_B → ADD_OUT)
ADD_OUT → Rd

PC[31:28] → ||_A
IMEM[25,0] || 02 → ||_B
||_OUT → MUX

MUX_OUT → PC
```

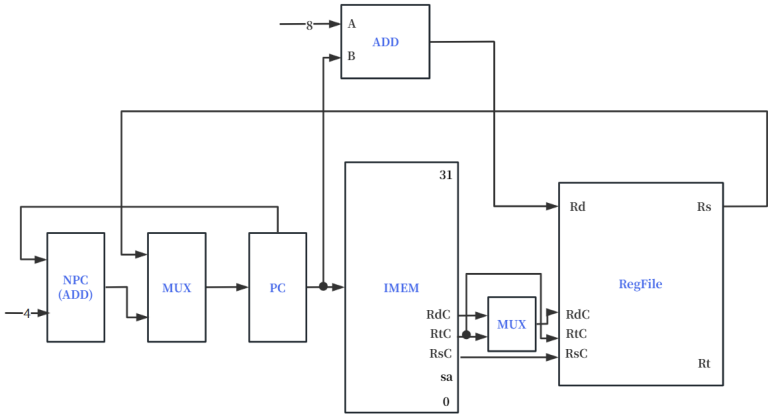
54条指令CPU

32.CLZ



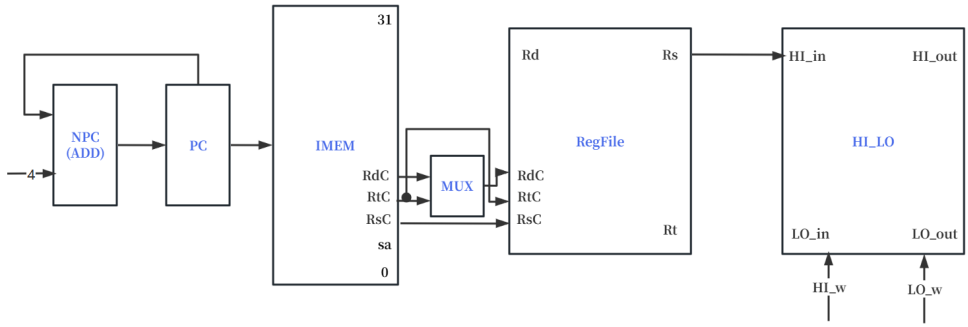

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → CLZ_in
CLZ_out → Rd
```

33.JALR



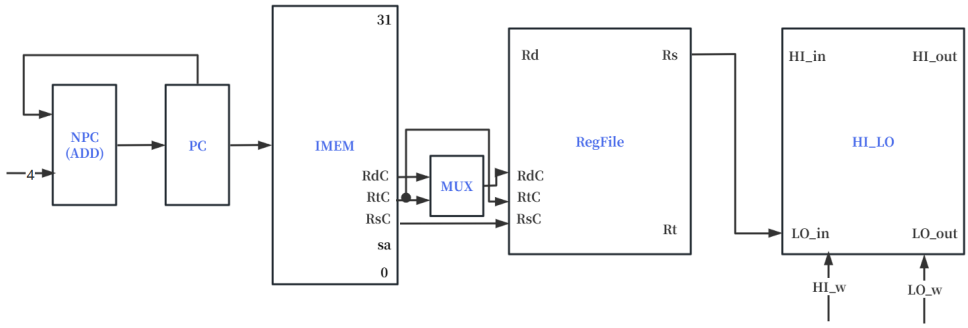
```
PC → IMEM
PC → ADD_A
ADD_A + ADD_B → ADD_out

PC + 4 → NPC
NPC → MUX

Rs → MUX
MUX(Rs) → PC
ADD_out → Rd
```

34.MTHI



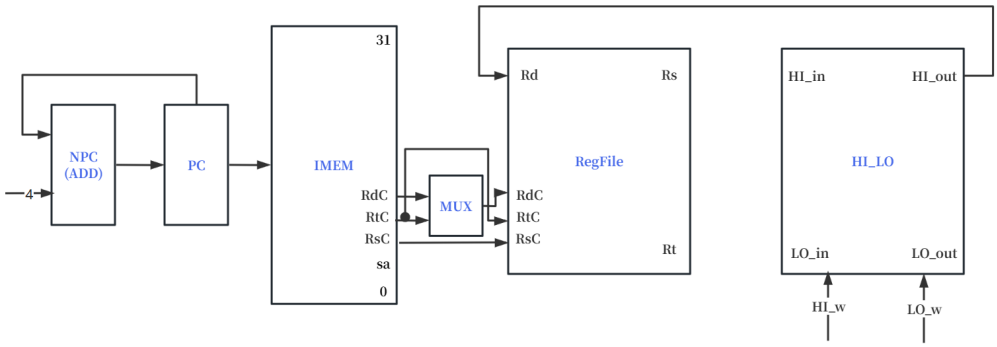

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs —HI_w→ HI_in
```

35.MTLO



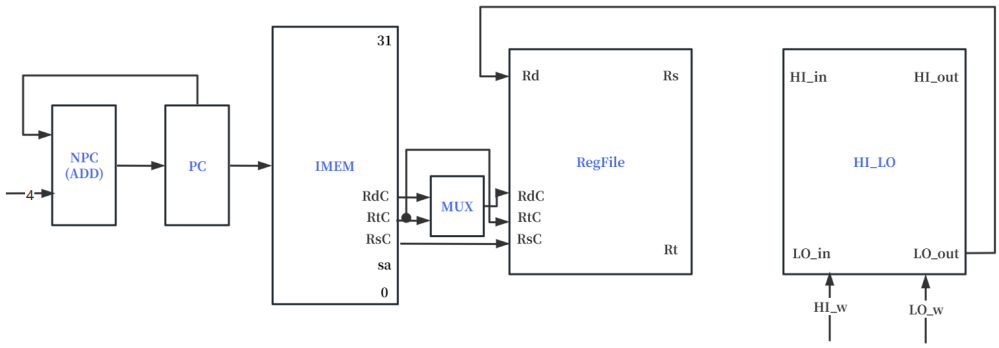

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs —LO_w→ HI_in
```

36.MFHI



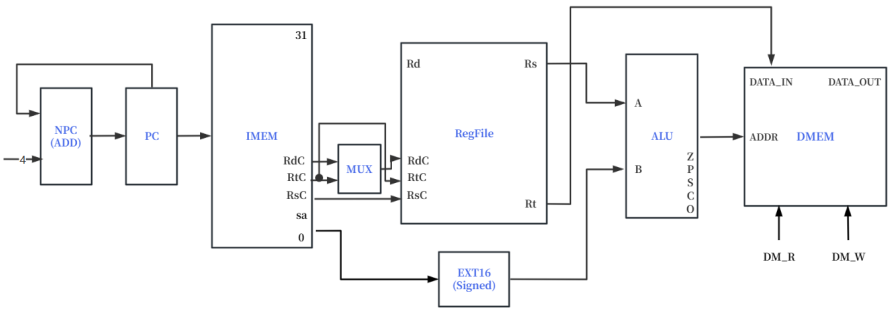

```
PC → IMEM
PC + 4 → NPC
NPC → PC

HI_out → Rd
```

37.MFLO



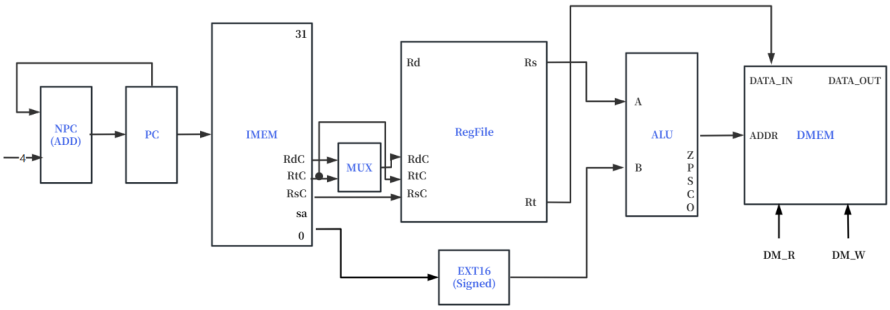

```
PC → IMEM
PC + 4 → NPC
NPC → PC

LO_out → Rd
```

38.SB



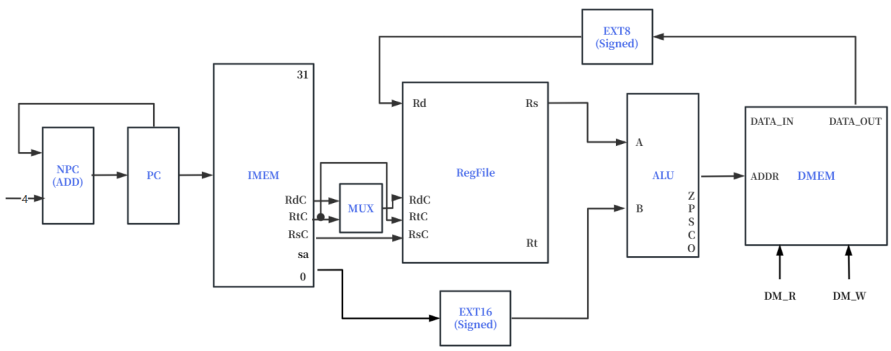
```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

ALU_out → DM_ADDR
Rt[7:0] —DM_W→ DATA_IN
```

39.SH




```
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

ALU_out → DM_ADDR
Rt[15:0] —DM_W→ DATA_IN
```

40.LB

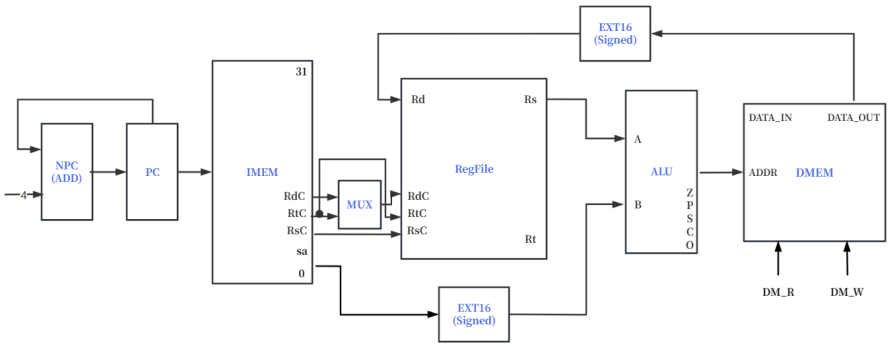


注：做了调整，改为Rd写入
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

ALU_out → DM_ADDR
DATA_OUT —DM_R→ EXT8_in
EXT8_out → Rd

41.LH

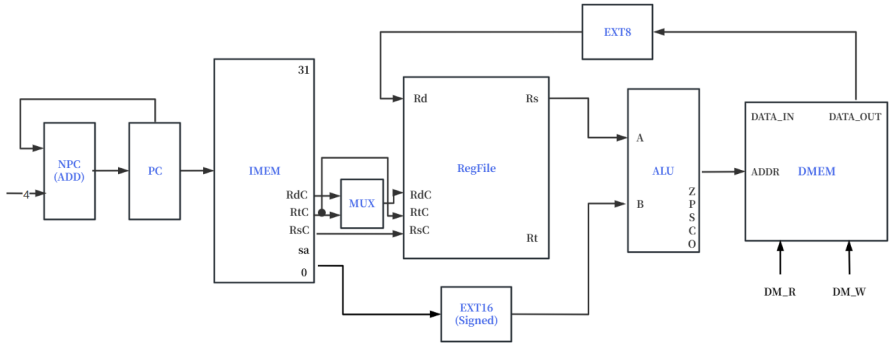


注：做了调整，改为Rd写入
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

ALU_out → DM_ADDR
DATA_OUT —DM_R→ EXT16_in
EXT16_out → Rd

42.LBU

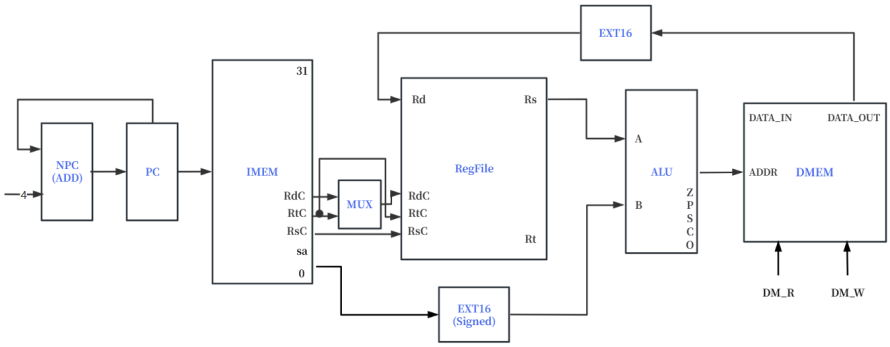


注：做了调整，改为Rd写入
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

ALU_out → DM_ADDR
DATA_OUT —DM_R→ EXT8_in
EXT8_out → Rd

43.LHU

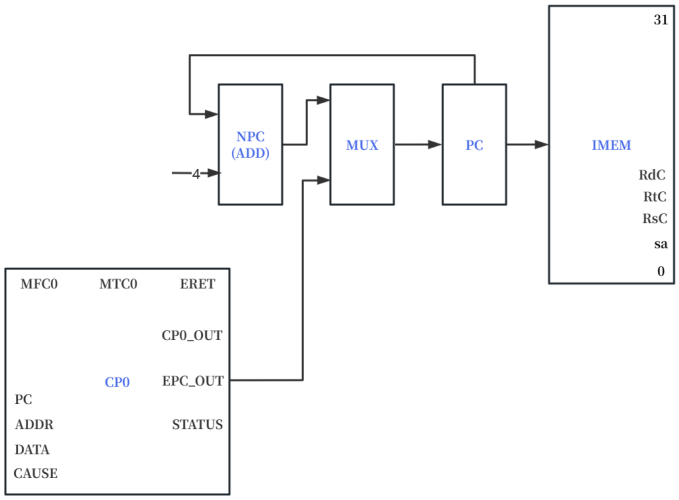


注：做了调整，改为Rd写入
PC → IMEM
PC + 4 → NPC
NPC → PC

IMEM[15:0] → EXT16_in
Rs → ALU_A
EXT16_out → ALU_B

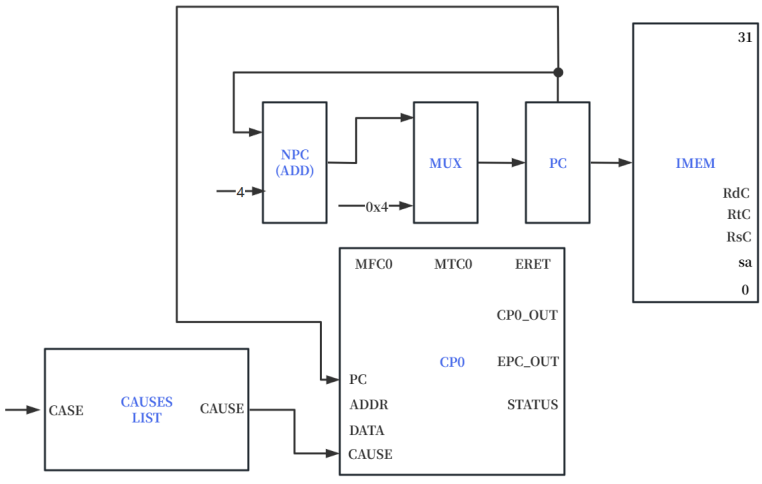
ALU_out → DM_ADDR
DATA_OUT —DM_R→ EXT16_in
EXT16_out → Rd

44.ERET



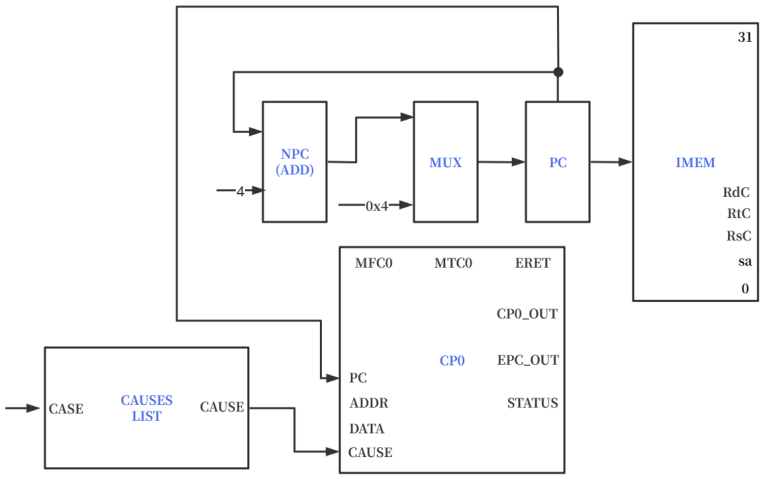

```
PC → IMEM
PC + 4 → NPC
NPC → MUX

(STAUTS >> 5 → STATUS) # CP0自己做
EPC_OUT → MUX
MUX(EPC_OUT) → PC
```

45.BREAK



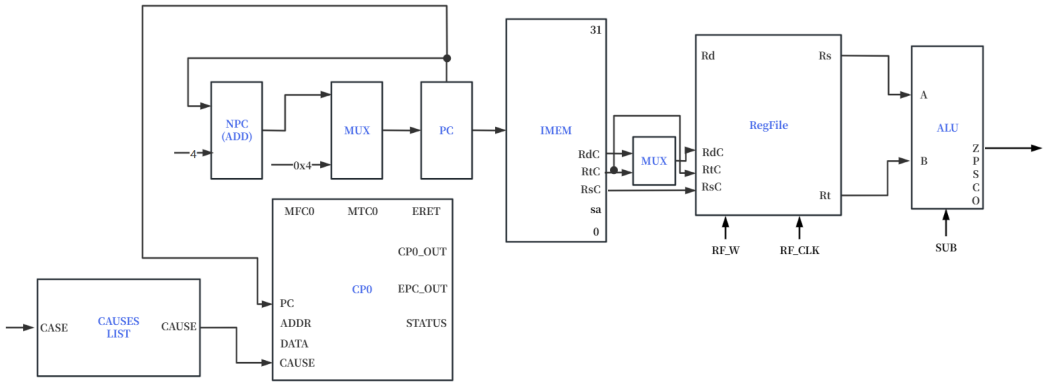

```
PC → IMEM
PC + 4 → NPC
NPC → MUX

PC → CP0
CAUSES-LIST_CAUSE → CP0_CAUSE
(STATUS << 5 → STATUS)

0x4 → MUX
MUX(0x4) → PC
```

46.SYSCALL




```
PC → IMEM
PC + 4 → NPC
NPC → MUX

PC → CP0
CAUSES-LIST_CAUSE → CP0_CAUSE
(STATUS << 5 → STATUS)

0x4 → MUX
MUX(0x4) → PC
```

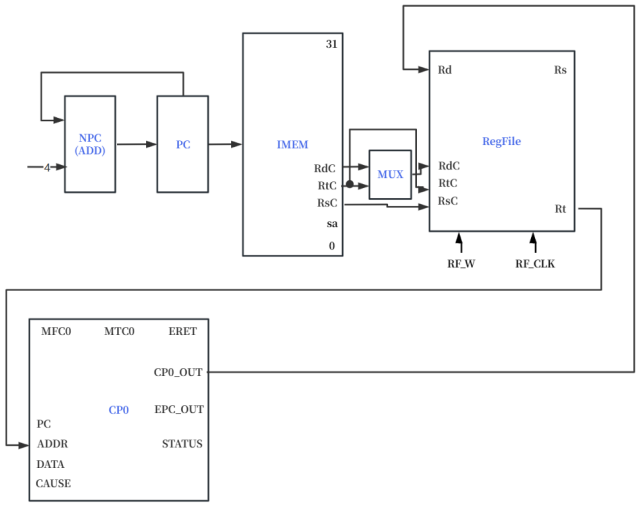
47.TEQ




```
PC → IMEM
PC + 4 → NPC
NPC → MUX
0x4 → MUX

Rs → ALU_A
Rt → ALU_B
ZERO → Rs - Rt == 0

if ZERO:
    PC → CP0
```

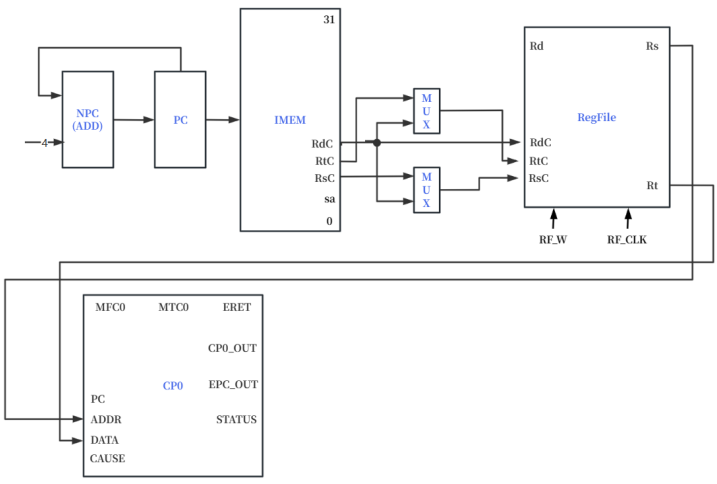
```
    CAUSES-LIST_CAUSE → CP0_CAUSE
    (STATUS << 5 → STATUS)

    MUX(0x4) → PC

else:
    MUX(NPC) → PC
```

48.MFC0



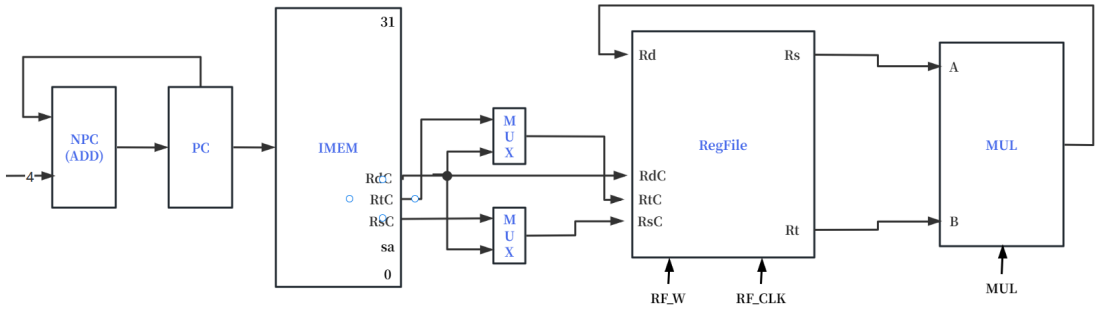

```
注：做了调整，互换了Rt和Rd，改为Rt输出地址，Rd写入
PC → IMEM
PC + 4 → NPC
NPC → PC

Rt → ADDR
CP0_OUT → Rd
```

49.MTC0



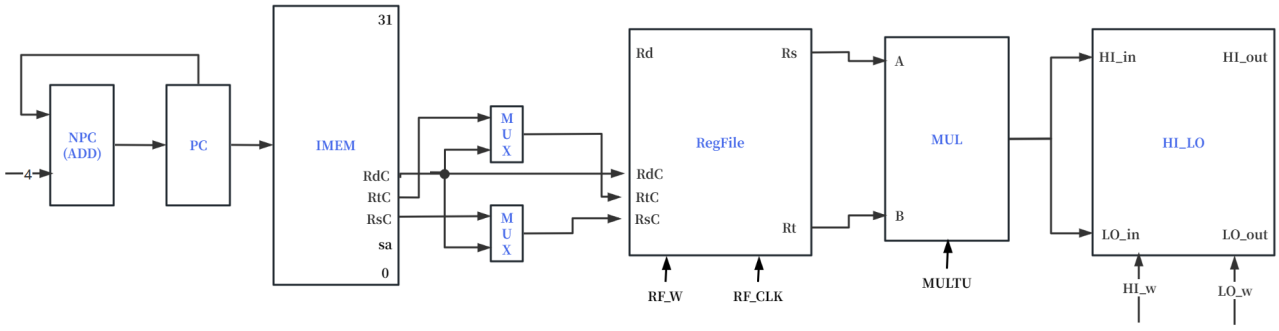
```
注：做了调整，互换了Rs和Rd，改为Rt输出要写入的值，Rs输出要写入的地址
PC → IMEM
PC + 4 → NPC
NPC → PC

Rt → ADDR
Rs → CP0_DATA
```

50.MUL



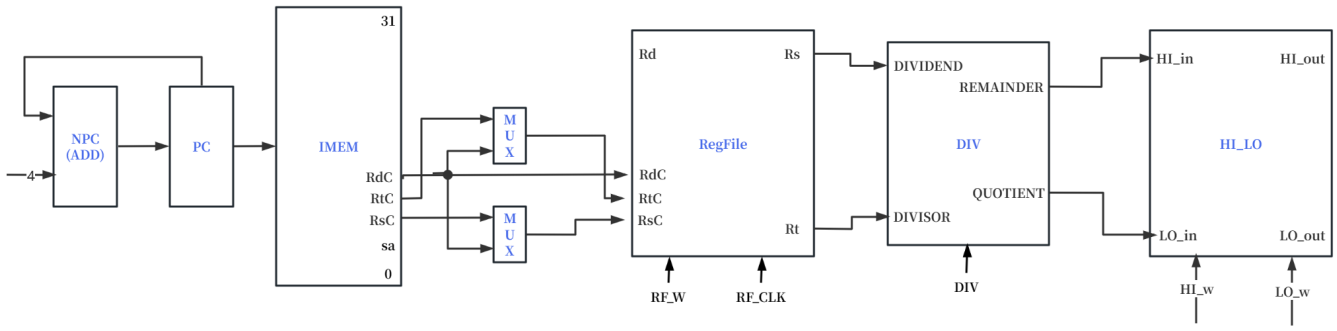

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → MUL_A
Rt → MUL_B
MUL_A * MUL_B → res
res[31:0] → Rd
```

51.MULTU



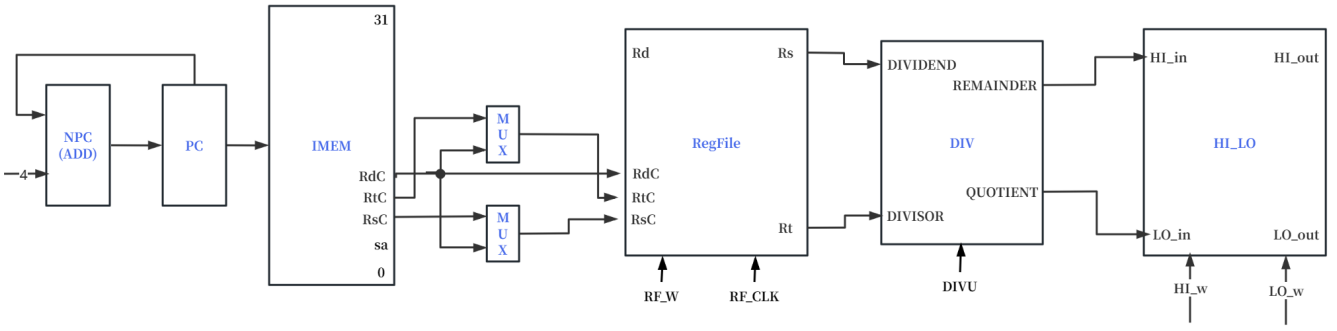

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → MUL_A
Rt → MUL_B
MUL_A * MUL_B → res
res[31:0] → LO_in
res[63:32] → HI_in
```

52.DIV



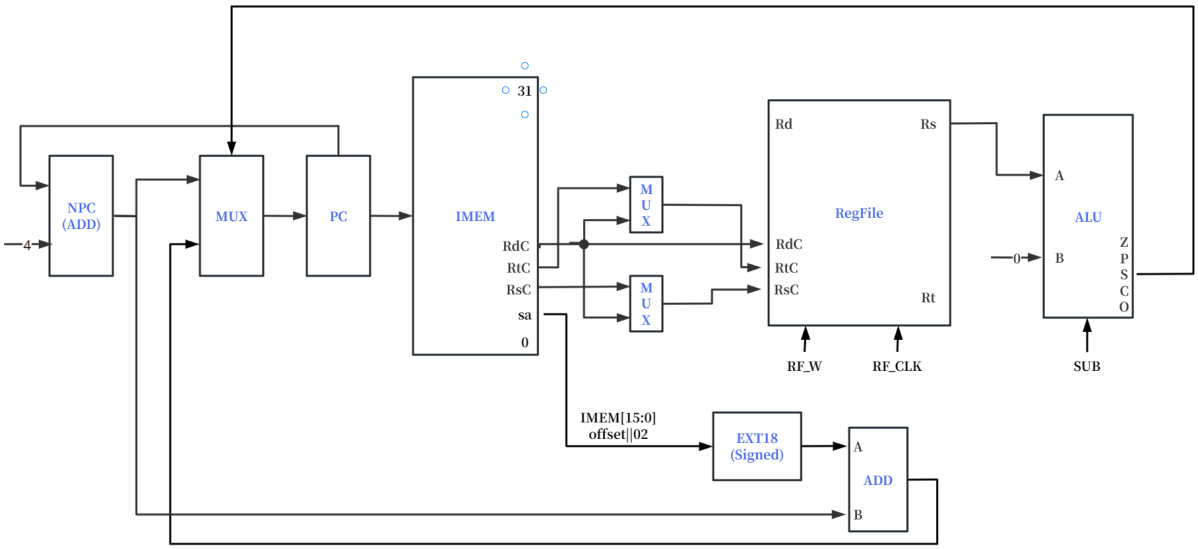

```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → DIVIDEND
Rt → DIVISOR
Rs/Rt → QUOTIENT
Rs%Rt → REMAINDER
QUOTIENT → HI_in
REMAINDER → LO_in
```

53.DIVU




```
PC → IMEM
PC + 4 → NPC
NPC → PC

Rs → DIVIDEND
Rt → DIVISOR
Rs/Rt → QUOTIENT
Rs%Rt → REMAINDER
QUOTIENT → HI_in
REMAINDER → LO_in
```

54.BGEZ




```
PC → IMEM
PC + 4 → NPC
NPC → MUX

Rs → ALU_A
0 → ALU_B
Rs - 0 → res

IMEM[15:0] || 02 → EXT18_in
EXT18_out → ADD_A
NPC → ADD_B
ADD_A + ADD_B → ADD_out

if S < 0
    MUX(NPC) → PC
else
    MUX(ADD_out) → PC
```