

## 本节主题



# 分支指令的 控制信号

北京大学·慕课  
计算机组成  
制作人：陆俊林





# 不同维度的指令分类

运算指令	addu rd,rs,rt subu rd,rs,rt	ori rt,rs,imm16	
访存指令		lw rt,imm16(rs) sw rt,imm16(rs)	
分支指令		beq rs,rt,imm16	
	R型指令	I型指令	J型指令

# 条件分支指令的示例



```
if (i==j)
    f=g+h;
else
    f=g-h;
```

C语言代码

```
beq $s3,$s4,True      # branch i==j
sub $s0,$s1,$s2        # f=g-h(false)
j   Next               # goto Next

True: add $s0,$s1,$s2 # f=g+h (true)
Next: ...
```

MIPS汇编语言代码

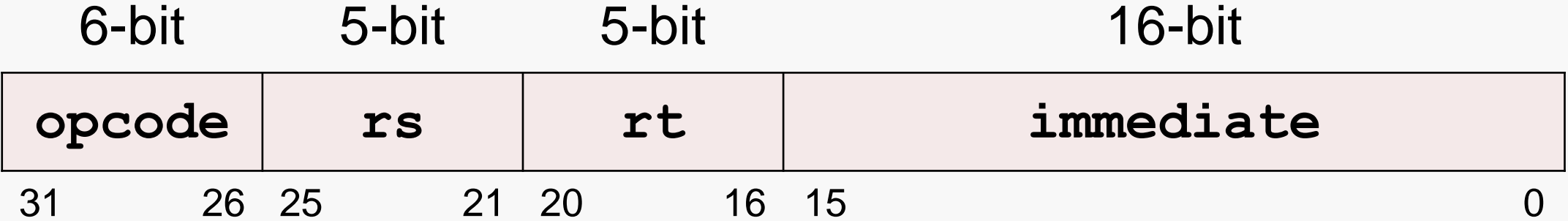


# beq指令的操作步骤

beq rs, rt, imm16

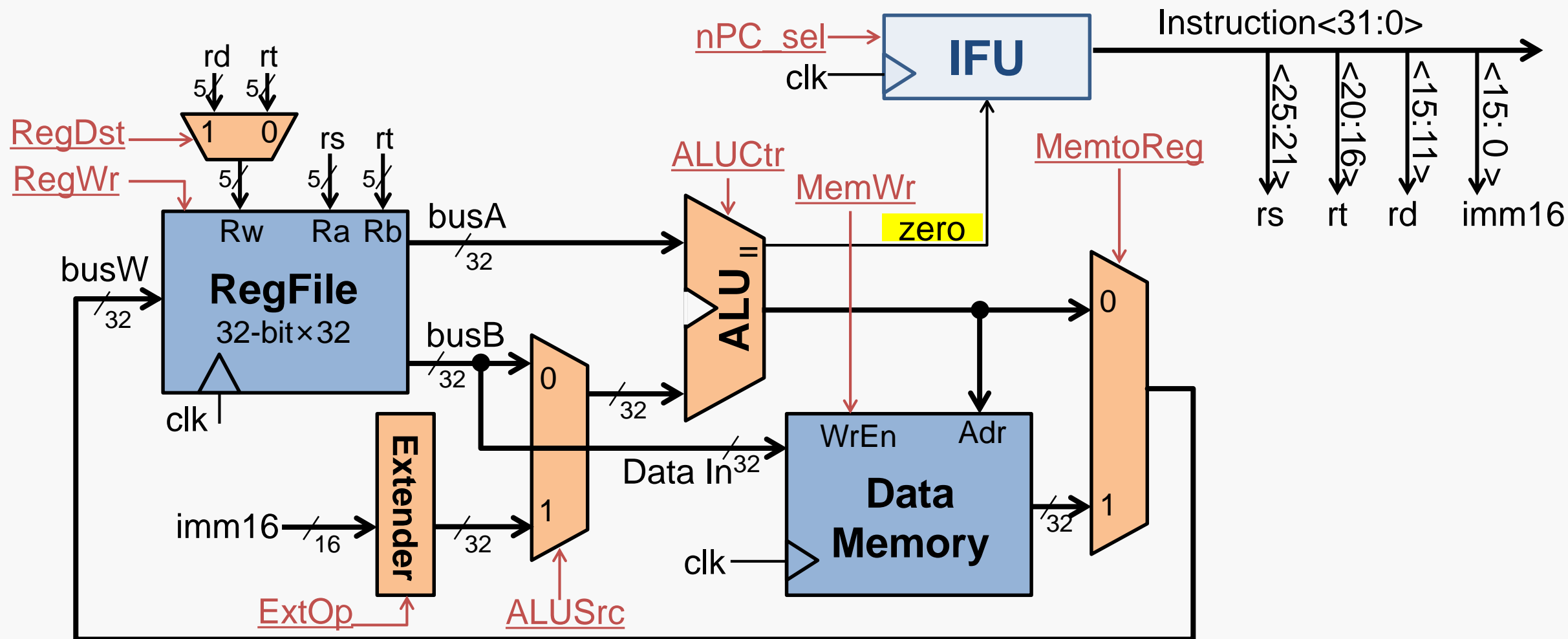
- ① MEM[PC] 从指令存储器中取回指令
- ② if (R[rs] - R[rt] == 0) 判断转移条件是否成立
- ③ then PC = PC + 4 + SignExt[imm16] \* 4 ;  
else PC = PC + 4 ;  
计算下一条指令的地址

I



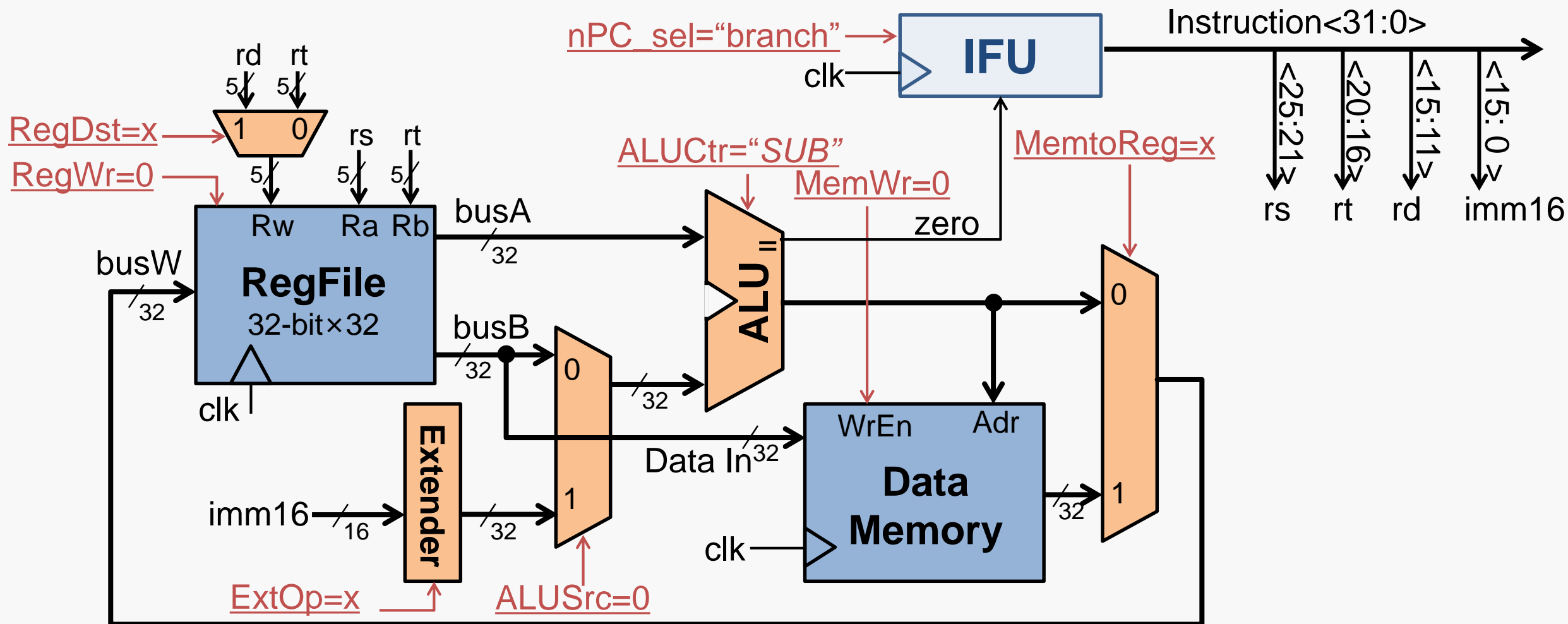
## beq指令的操作步骤（2）

if ( $R[rs] - R[rt] == 0$ ) then zero=1; else zero=0



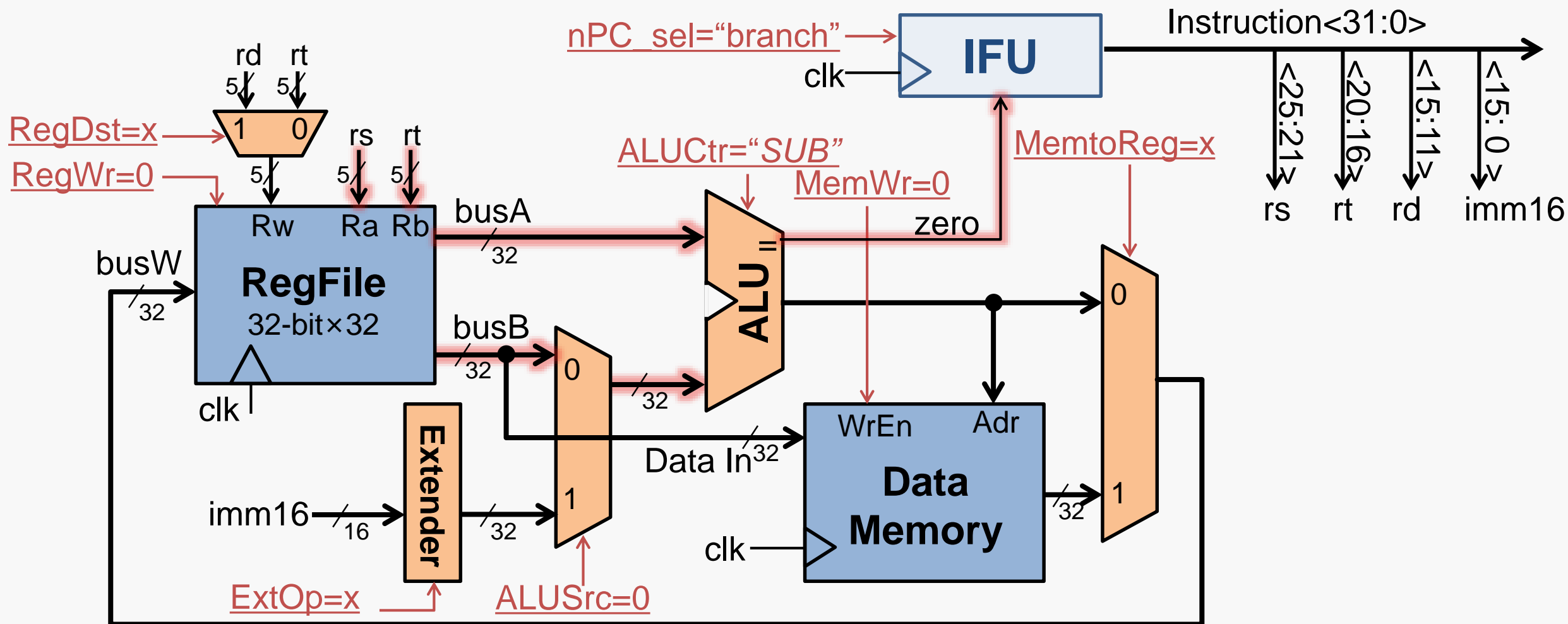
## beq指令的操作步骤（2）

if ( $R[rs] - R[rt] == 0$ ) then zero=1; else zero=0



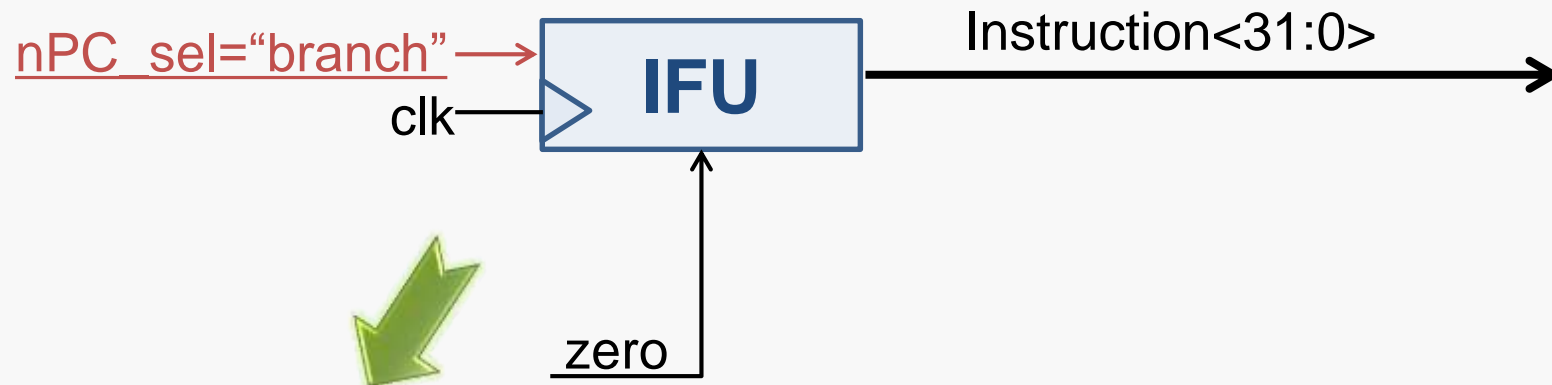
## beq指令的操作步骤（2）

if ( $R[rs] - R[rt] == 0$ ) then zero=1; else zero=0

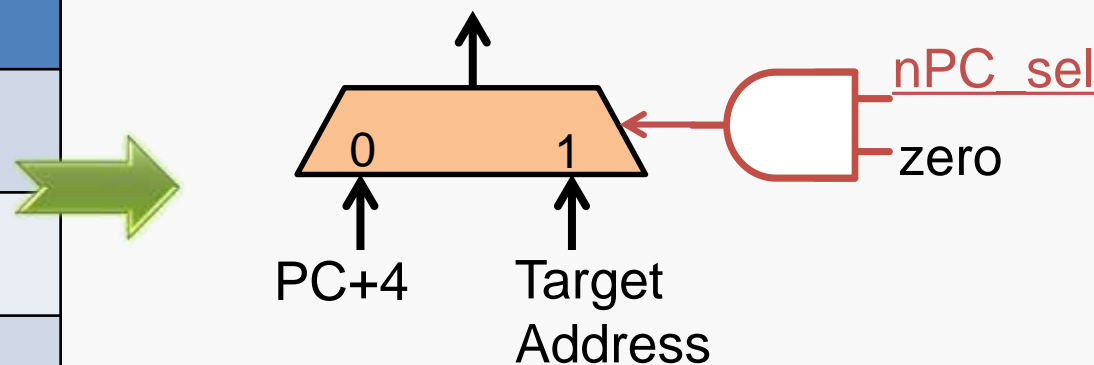


## beq指令的操作步骤（3）

```
if (zero==0) then PC=PC+4 + SignExt[imm16]*4 ;  
else PC=PC+4;
```



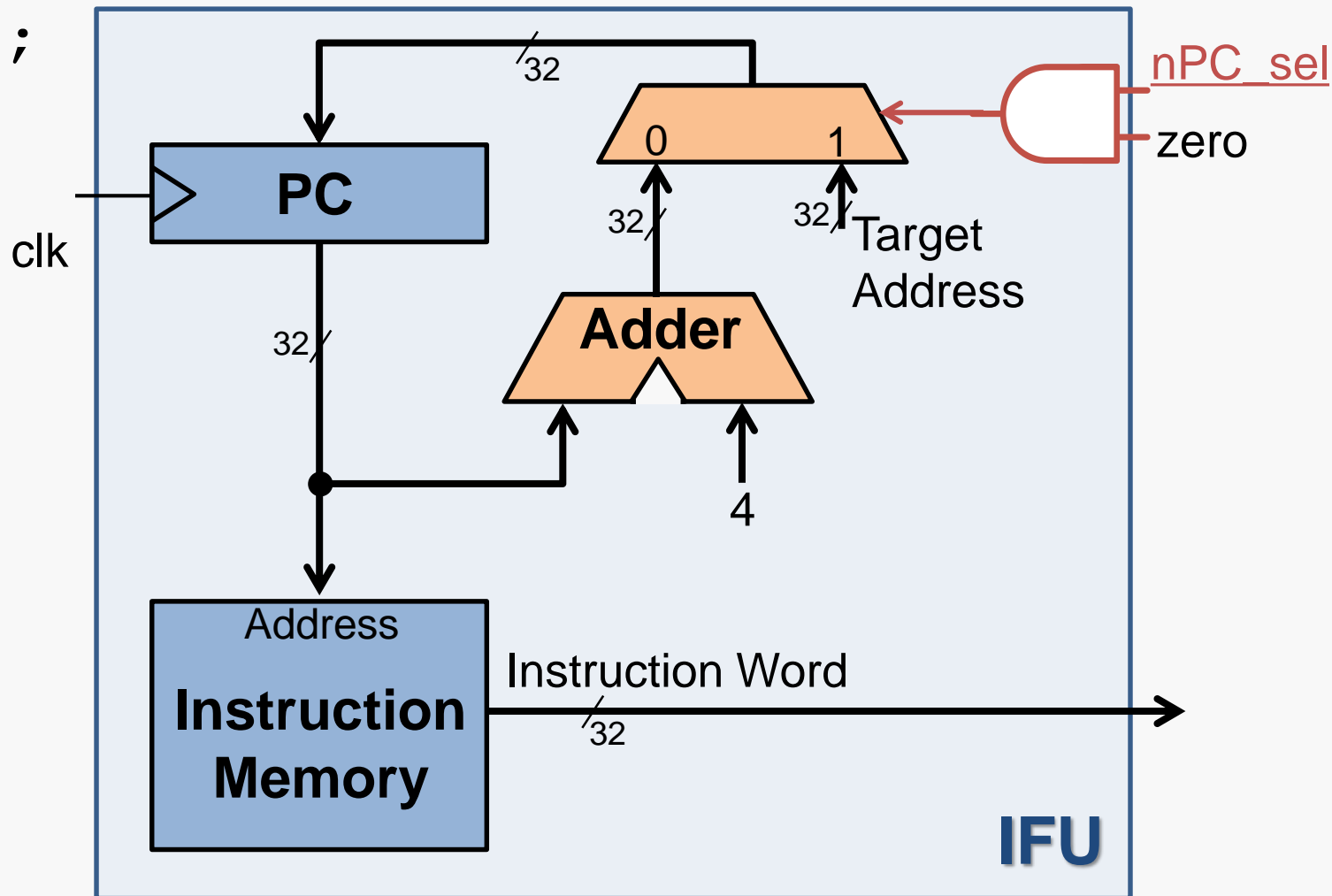
nPC_sel	zero	MUX
0 ( "+4" )	x	0 ( PC+4 )
1 ( "branch" )	0 ( ≠ )	0 ( PC+4 )
1 ( "branch" )	1 ( = )	1 ( Target Address )





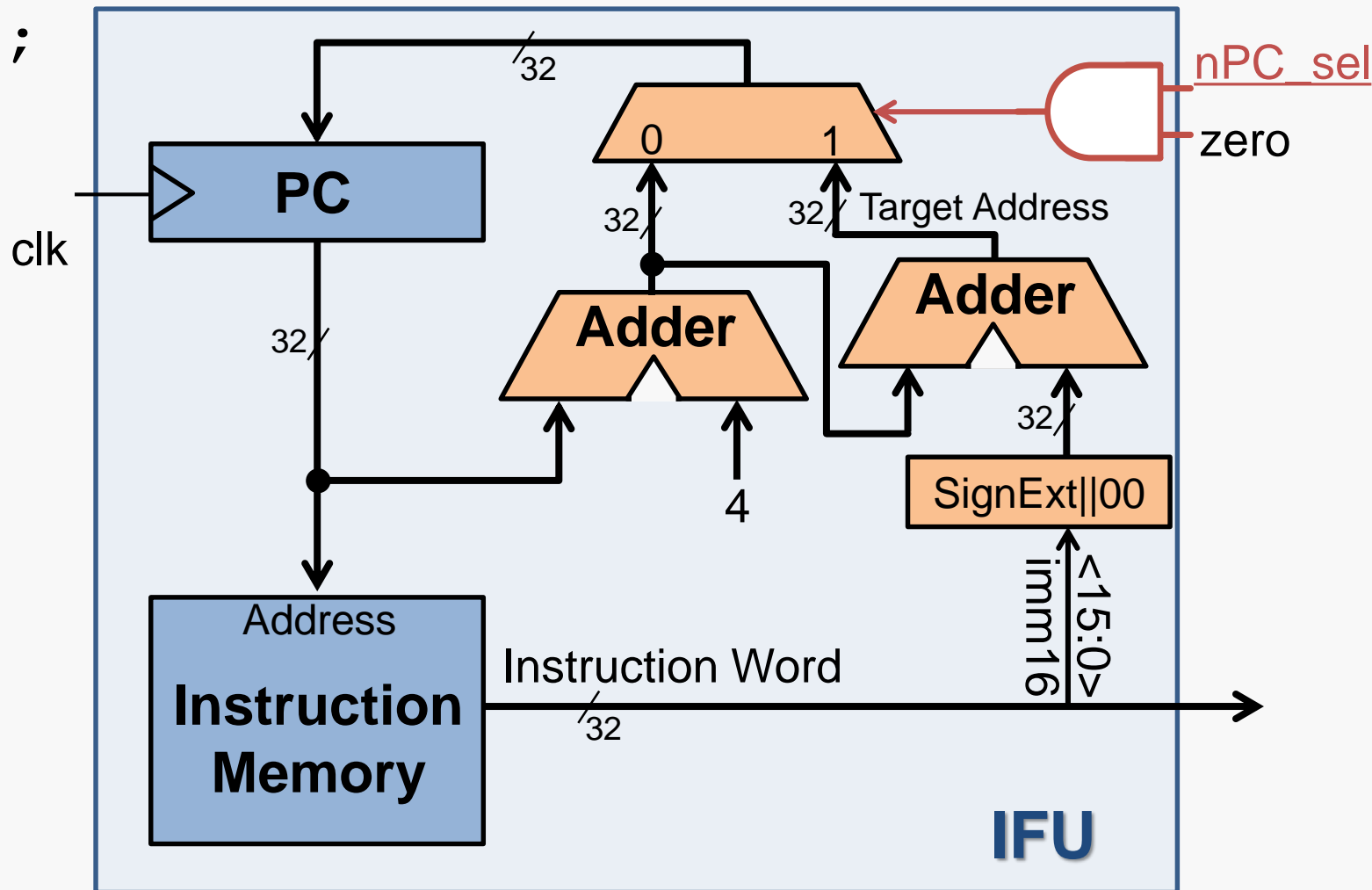
## beq指令的操作步骤（3）

`if (zero==0) then PC=PC+4 + SignExt[imm16]*4 ;`  
`else PC=PC+4;`



## beq指令的操作步骤（3）

if (zero==0) then  $PC = PC + 4 + \text{SignExt}[\text{imm16}] * 4$  ;  
else  $PC = PC + 4$  ;



## 本节小结



# 分支指令的 控制信号

北京大学·慕课  
计算机组成  
制作人：陆俊林

