

# 运算器部件组成及设计

2020年秋

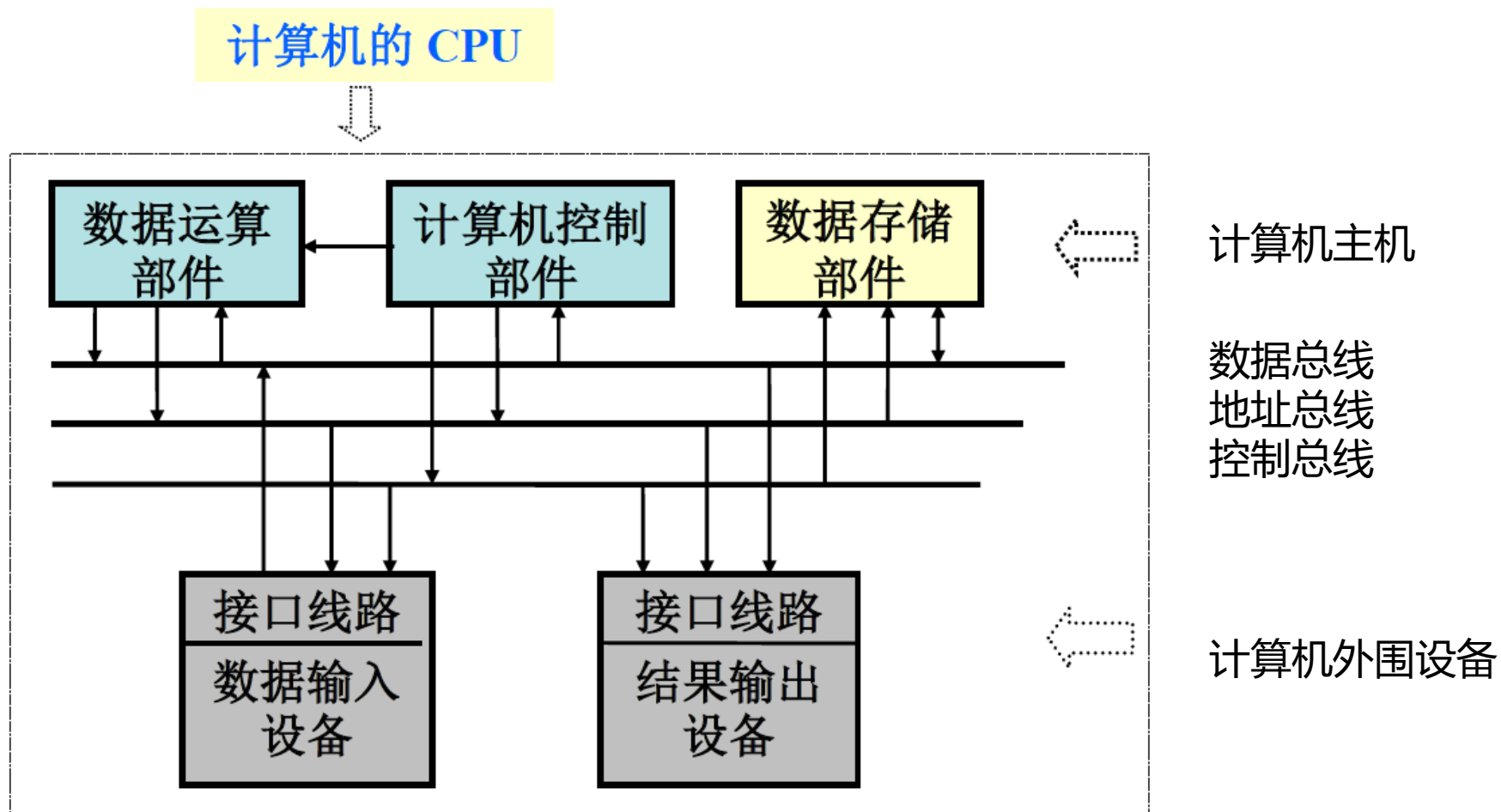
---

# 本讲概要

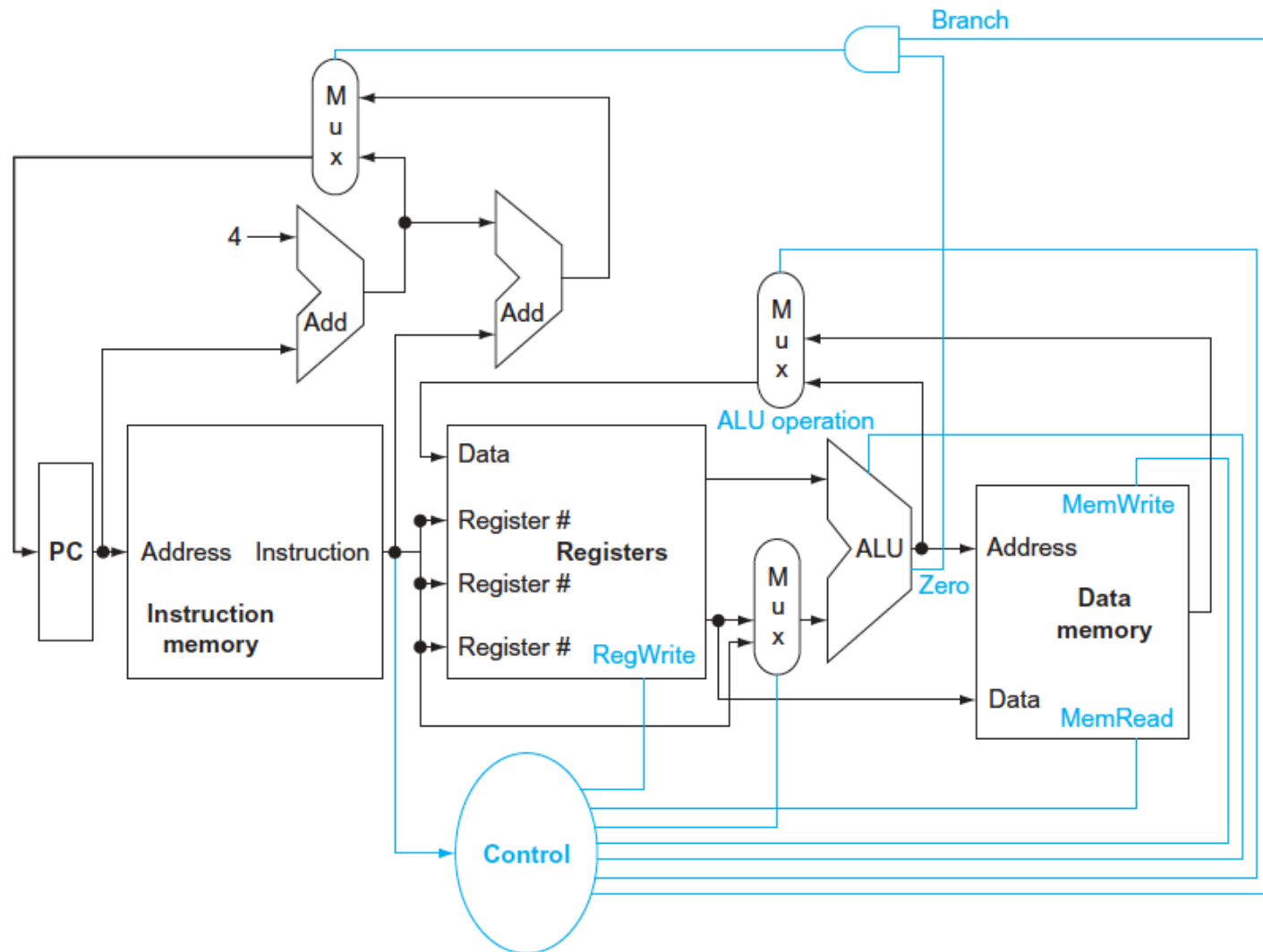
---

- ▶ 运算器功能与组成概述
- ▶ 熟悉Verilog语言
- ▶ 定点运算器实例AM2901
- ▶ 用Verilog语言描述AM2901

# 硬件系统的功能部件



# CPU示例



# 运算器的基本功能

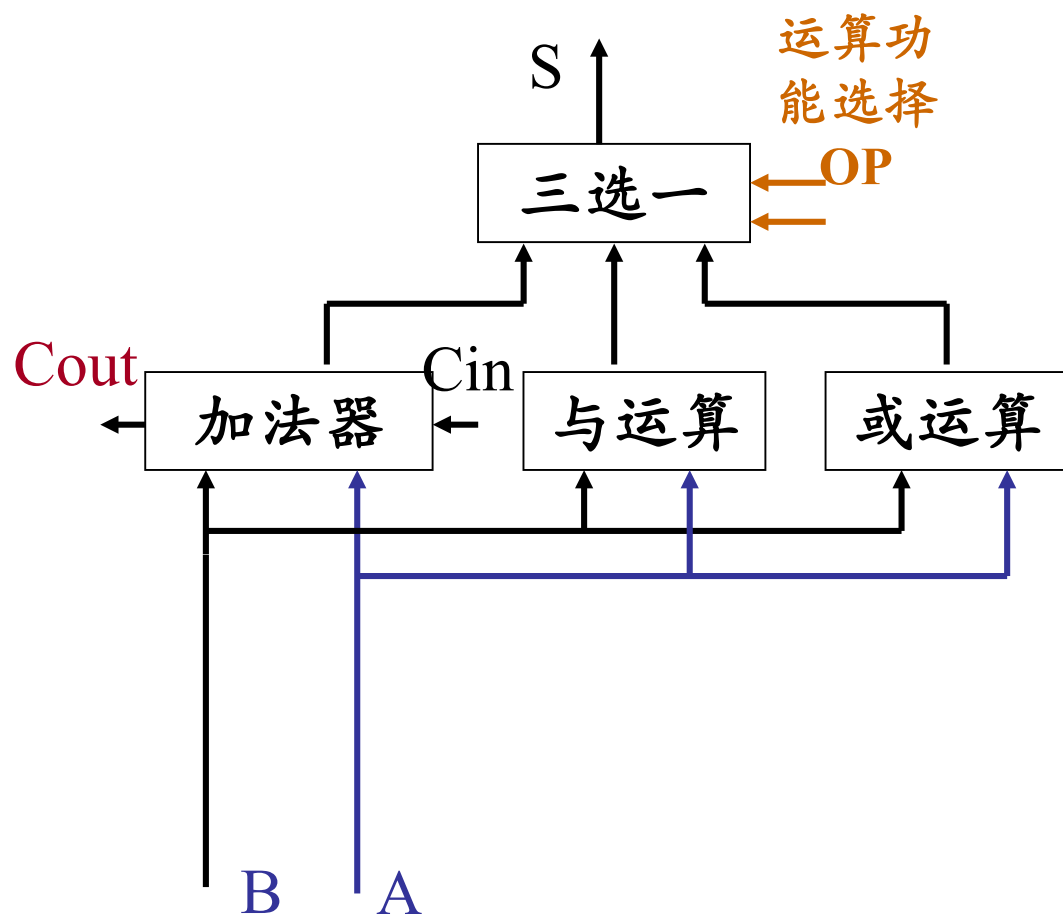
---

- ▶ 完成算术、逻辑运算，产生运算结果
  - ▶ ALU执行+、—、 $\times$ 、 $\div$ 、 $\wedge$ 、 $\vee$ 、 $\neg$
- ▶ 并给出运算结果的状态信息C,Z,V,S
- ▶ 暂存运算所用操作数
  - ▶ 寄存器组、立即数、数据总线
- ▶ 暂存运算的中间结果
  - ▶ 寄存器组、Q寄存器、移位线路
- ▶ 输出运算结果
  - ▶ 寄存器组、数据总线

运算器是计算机系统中执行数据运算、处理的功能部件，类似于一个工厂中的生产加工车间，由ALU和寄存器组等组成

# 设计支持 +、 $\wedge$ 、 $\vee$ 功能的 ALU 线路

一位的ALU的线路设计：1、首先画出其组成示意图



输入数据 A、B

加减进位输入 Cin

加减进位输出 Cout

3种运算用3部分电路

运算结果S, 3选1得到

运算功能选择控制

+/-选择 B 或 /B 的控制



# 设计支持 +、 $\wedge$ 、 $\vee$ 功能的 ALU 线路

一位的ALU的线路设计：2、接着写出功能的真值表

OP	A	B	Cin	S	Cout
00	0	0	0	0	0
00	0	1	0	1	0
00	1	0	0	1	0
00	1	1	0	0	1
00	0	0	1	1	0
00	0	1	1	0	1
00	1	0	1	0	1
00	1	1	1	1	1

(加法)

OP	A	B	S (与)
10	0	0	0
10	0	1	0
10	1	0	0
10	1	1	1

OP	A	B	S (或)
11	0	0	0
11	0	1	1
11	1	0	1
11	1	1	1



# 设计支持 +、∧、∨ 功能的 ALU 线路

一位的ALU的线路设计：3、按真值表写逻辑表达式

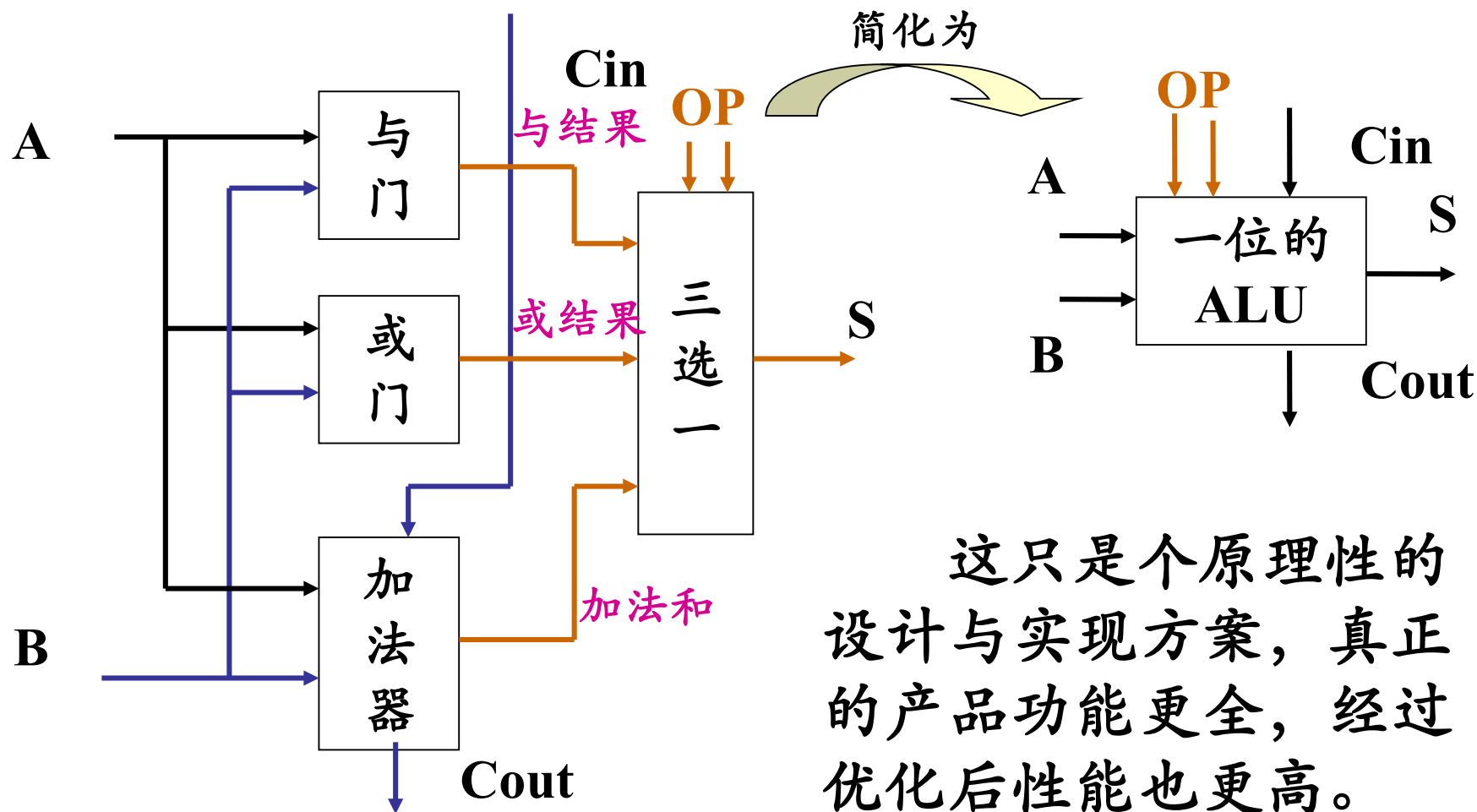
$$\begin{aligned} S = & (/OP1 \cdot /OP0) \cdot ( A \cdot /B \cdot /Cin \quad \text{加法运算} \\ & + /A \cdot B \cdot /Cin \\ & + /A \cdot /B \cdot Cin \\ & + A \cdot B \cdot Cin ) \\ & + ( OP1 \cdot /OP0 ) \cdot ( A \cdot B ) \quad \text{与运算} \\ & + ( OP1 \cdot OP0 ) \cdot ( A + B ) \quad \text{或运算} \end{aligned}$$

$$\begin{aligned} Cout = & (/OP1 \cdot /OP0) \cdot ( A \cdot B \quad \text{加法运算} \\ & + A \cdot Cin \\ & + B \cdot Cin ) \end{aligned}$$

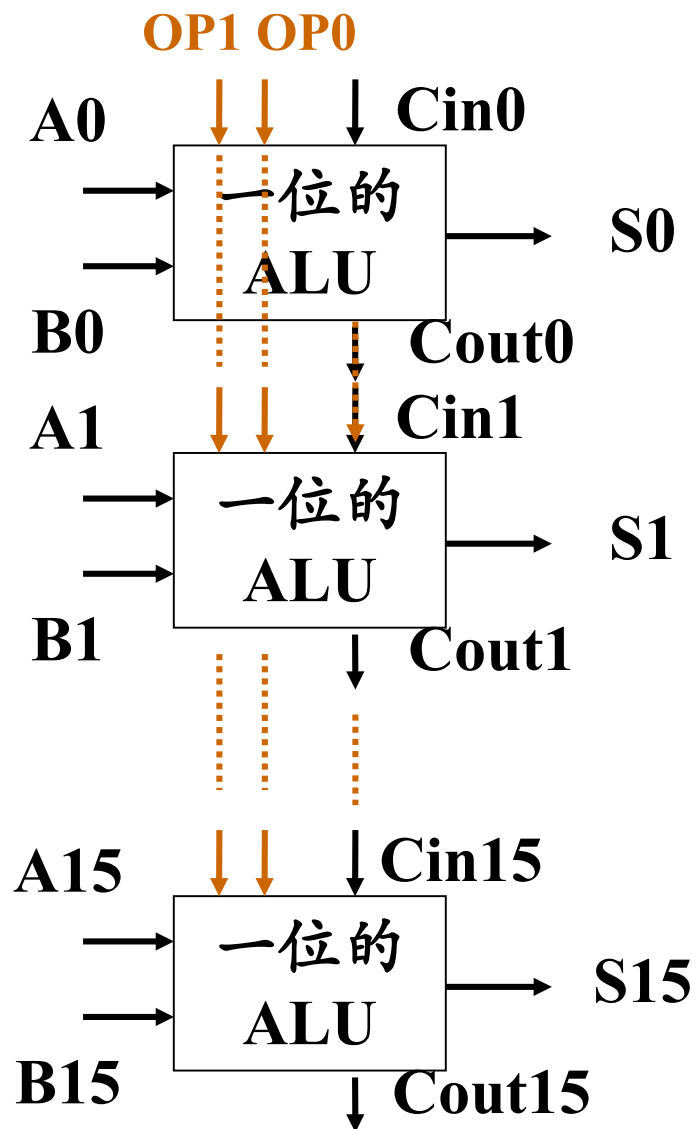


# 设计支持 +、 $\wedge$ 、 $\vee$ 功能的 ALU 线路

一位的ALU的线路设计：4、按逻辑表达式画出电路图



# 设计支持 +、 $\wedge$ 、 $\vee$ 功能的16位ALU



用 16 个一位的 ALU 实现一个 16 位的 ALU，使其并行执行算逻运算，使每位 ALU 处在不同的数位上，信号名加位序号。

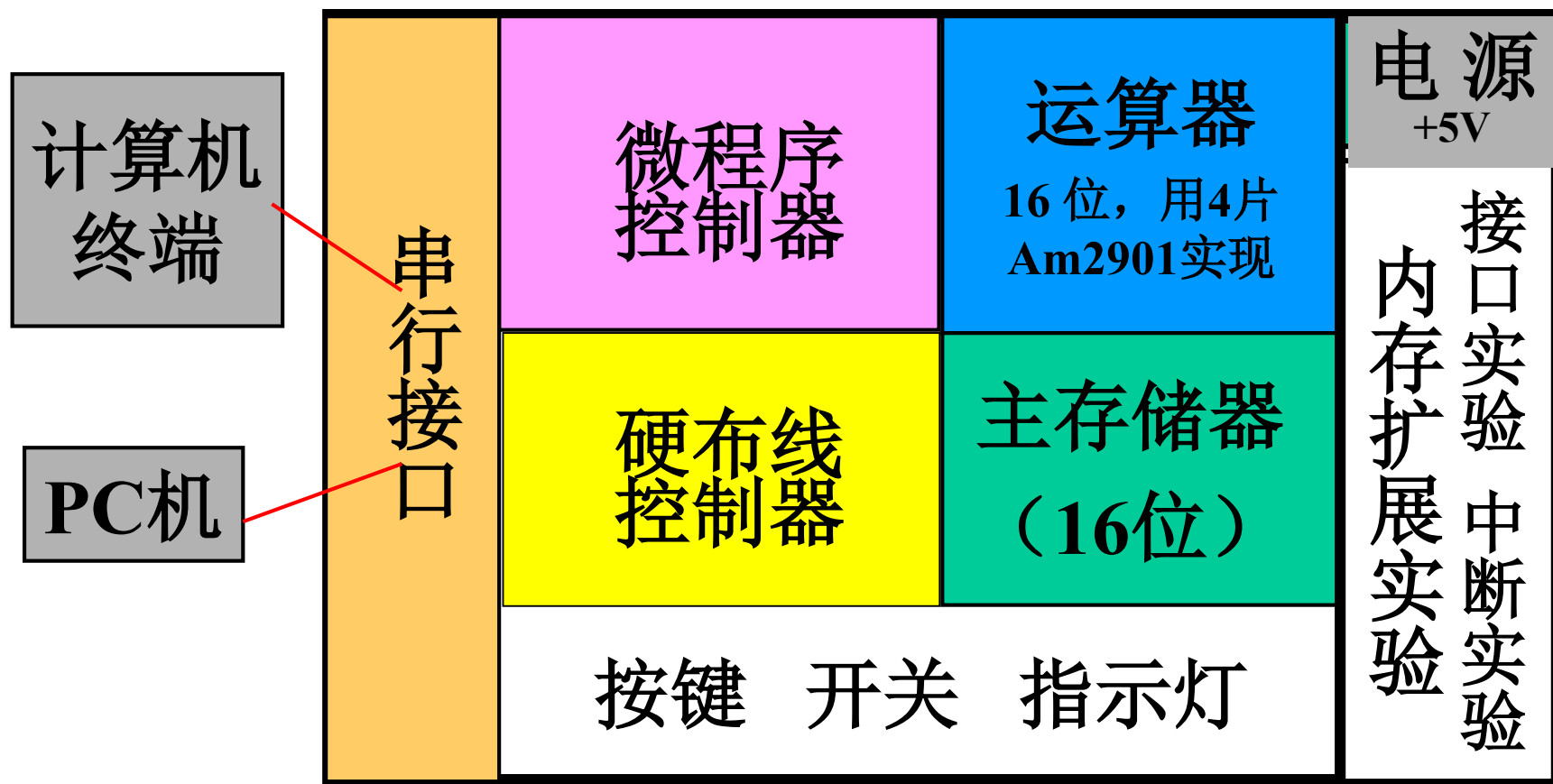
需要把每一位上的 ALU 的进位输出信号和相邻高位的进位输入信号正确连接；并把全部位上的控制信号连接在一起，使它们同步执行相同的运算功能，从而对两个 16 位的数据执行正确的运算功能，产生正确的结果。

还需要解决进位信号传送速度，可采用超前进位的方案。

# 简单的单周期 运算器AM2901

# TEC-2000?教学计算机硬件系统的组成

运算器用4片Am2901芯片实现



# TEC2000? 双CPU教学计算机系统

控制器部件  
(2 种类型)

运算器部件

存储器部件

串行口电路

微指令下地址

双 CPU  
教学计算机系统

5V  
直流电源

用VHDL描述功能，用FPGA器件实现的第2个CPU系统

地址

数据

ALU

IR

32位控制信号

Flags

节拍

按键、功能开关

36位控制信号开关

16位数据开关

辅助电路

这个系统的优点在于修改已有设计或增加新的指令等，主要表现为修改ABEL或VHDL语言的程序源码，编译后重新下载即可，受布线影响小，教学实验效率高



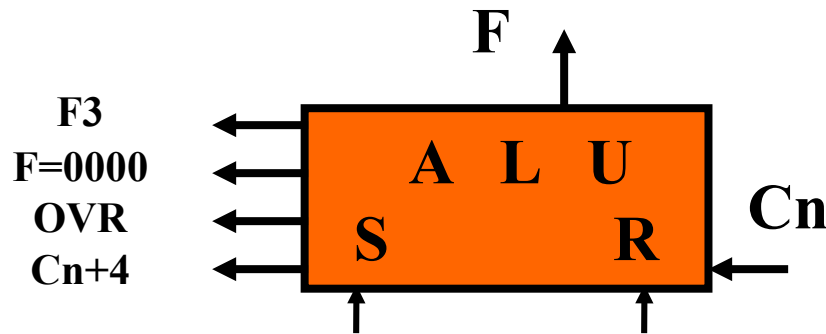
# 运算器功能与组成概述

- 计算机五大功能部件之一，在控制器的指挥控制下，完成指定给它的运算处理功能。
- 运算器通常包括定点运算器和浮点运算器两种类型：
  - **定点运算器**：完成对整数类型数据的算术运算、逻辑类型数据的逻辑运算
  - **浮点运算器**：完成对浮点类型数据的算术运算





符号位  
结果为零  
结果溢出  
进位输出



Am2901芯片是一个 4 位的位片结构的运算器器件,其内部组成如下:

第一个组成部分是算逻辑运算部件ALU,完成 3 种算术运算和 5 种逻辑运算,共计 8 种功能。

其输出为 F, 两路输入为 S、R, 最低位进位Cn, 4 个状态输出信号如图所示

### 3位功能选择码    8种运算功能

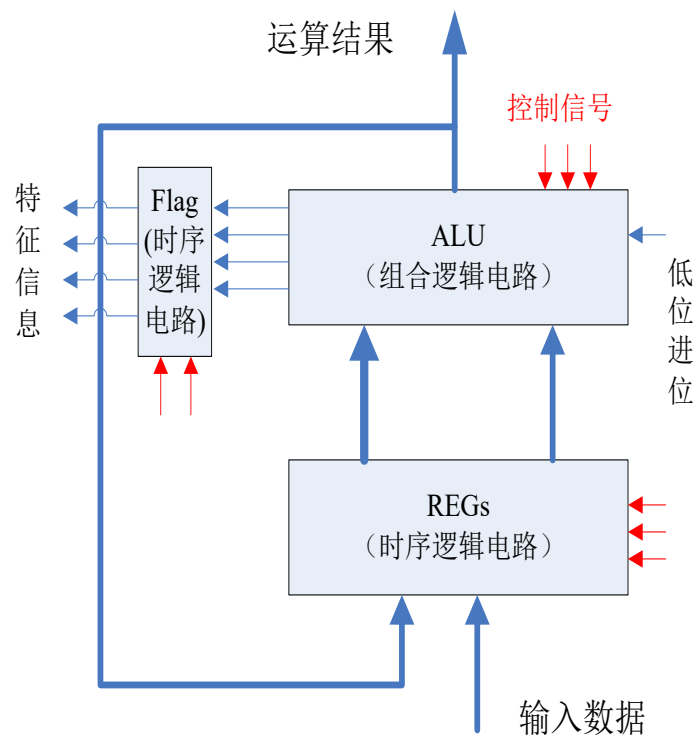
000	$R+S$
001	$S-R$
010	$R-S$
011	$R \vee S$
100	$R \wedge S$
101	$\overline{R} \wedge S$
110	$R \oplus S$
111	$\overline{R \oplus S}$

# 运算器设计需要解决的问题

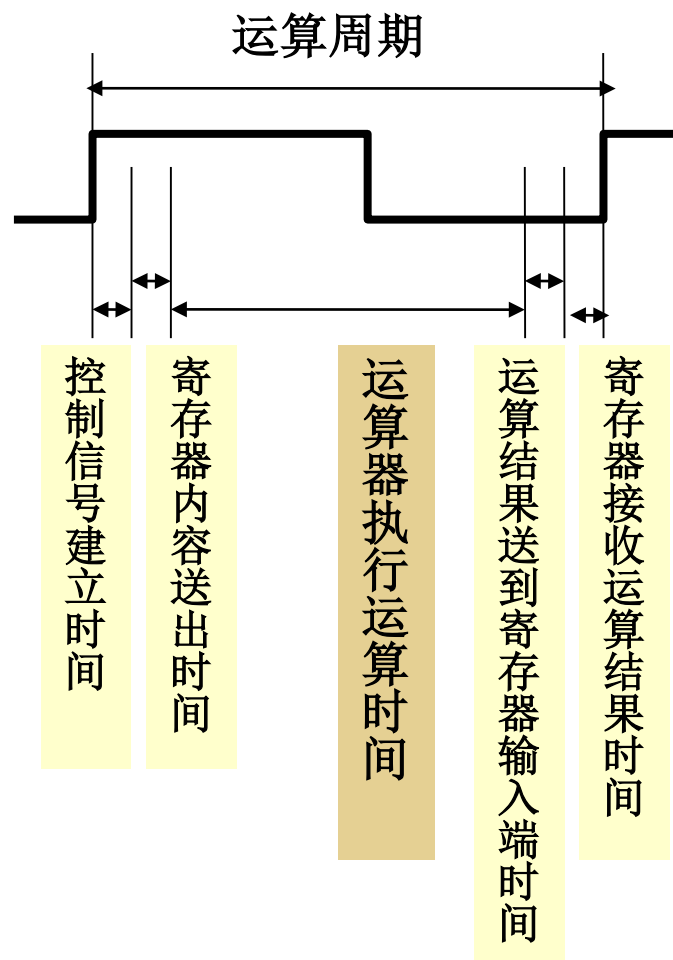
- 需要明确参加运算的数据来源，运算结果的去向。运算器能直接运算的数据，通常来自于运算器本身的寄存器。这些寄存器本身是暂存数据用的，是由触发器构成的时序逻辑电路。
- 需要明确将要执行的运算功能，是对数值数据的何种算术运算功能，还是对逻辑数据的何种逻辑运算功能。完成数据运算功能的线路是组合逻辑电路。
- 运算器完成一次数据运算过程由多个时间段组成，其时序关系示意表示在下图。
- 运算器部件只有和计算机的其他部件连接起来才能协同完成指令的执行过程。



# 单步完成的一次运算的时序关系



运算器组成示意图

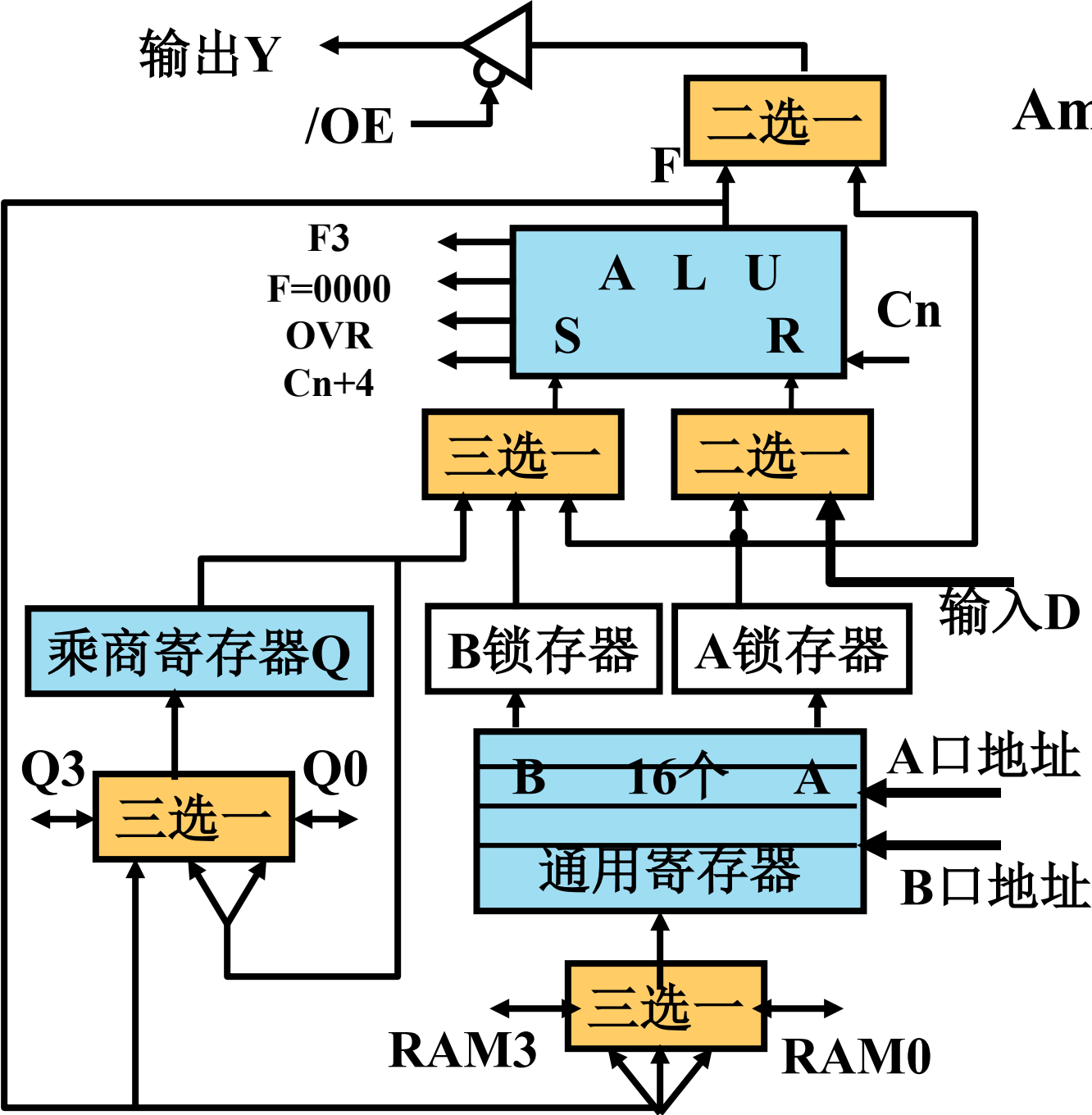


一个运算周期中的各时间段示意图

# 定点运算器功能与组成

- 完成算术与逻辑运算功能
    - 算术逻辑单元（ALU）
  - 暂存参加运算的数据和中间结果
    - 通用寄存器组
  - 乘除法运算的硬件线路支持
    - 乘商寄存器（Q寄存器）
- 通过几组多路选择器电路实现相互连接，以便数据传送。
- 作为处理机内部数据通路（Data Path）

# Am2901内部组成

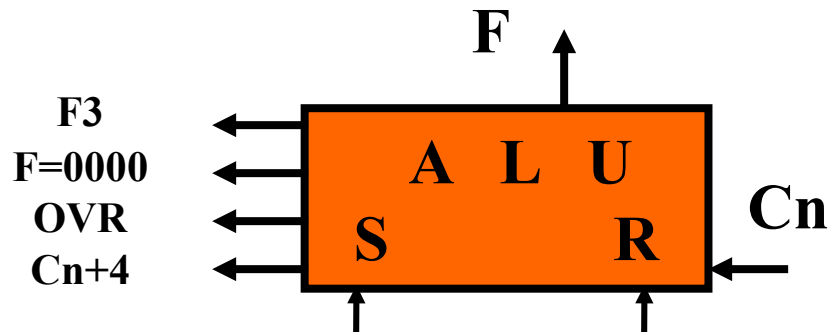


- 组成
- 算逻运算部件
  - 通用寄存器组
  - 乘商寄存器 Q

- 功能
- 8种运算功能
  - 8种数据组合
  - 8种结果处理



符号位  
结果为零  
结果溢出  
进位输出



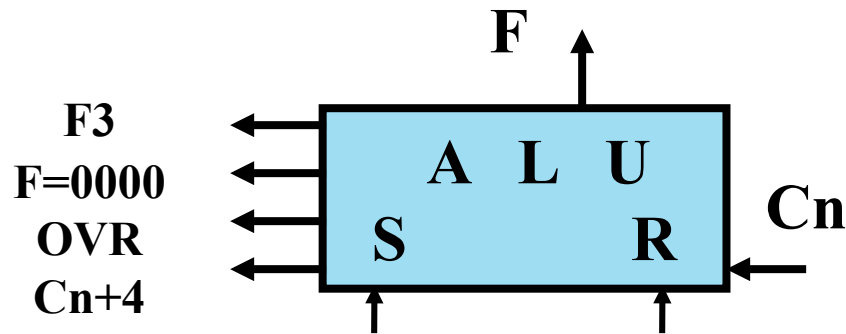
Am2901芯片是一个 4 位的位片结构的运算器器件,其内部组成如下:

第一个组成部分是算逻辑运算部件ALU,完成 3 种算术运算和 5 种逻辑运算,共计 8 种功能。

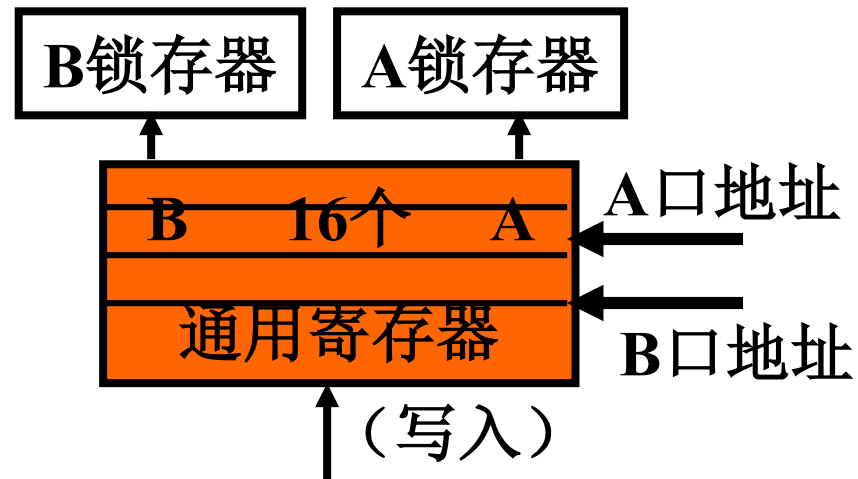
其输出为 F, 两路输入为 S、R, 最低位进位Cn, 4个状态输出信号如图所示

### 3位功能选择码      8种运算功能

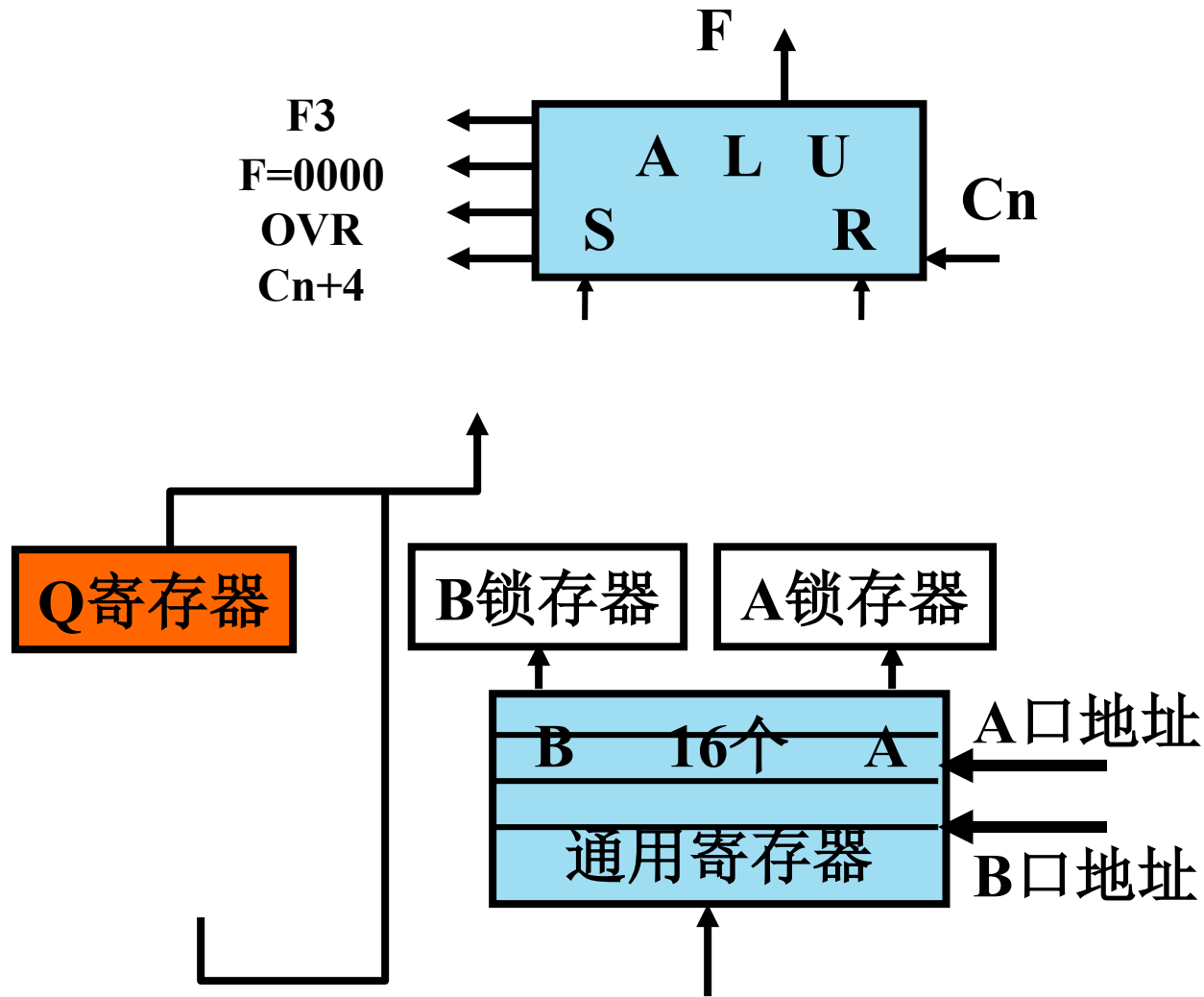
000	$R+S$
001	$S-R$
010	$R-S$
011	$R \vee S$
100	$R \wedge S$
101	$\overline{R} \wedge S$
110	$R \oplus S$
111	$\overline{R \oplus S}$



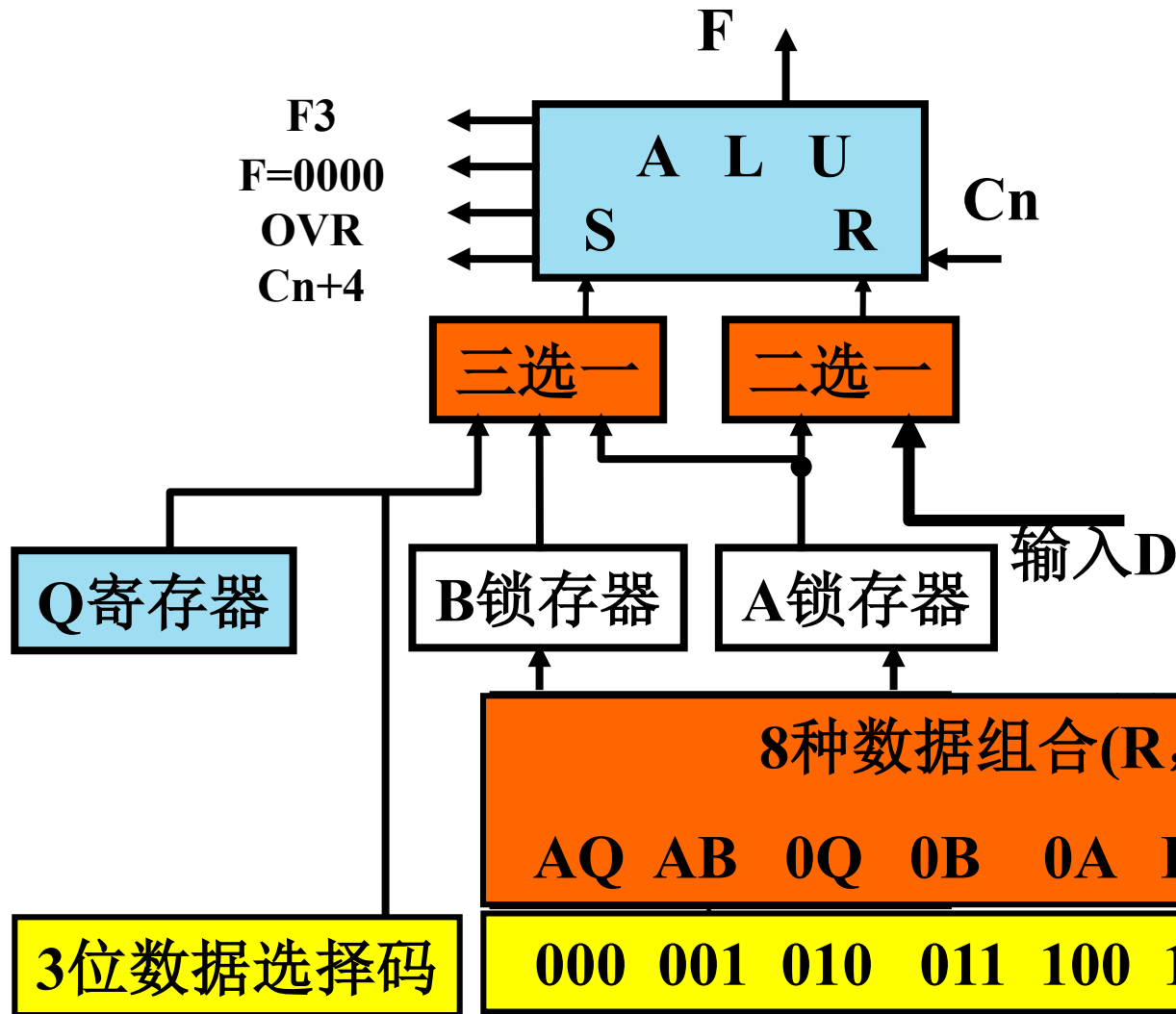
第二个组成部分是通用寄存器组由16个寄存器构成，并通过B口与A口地址选择被读的寄存器，B口地址还用于指定写入寄存器



通过B口地址、A口地址读出的数据将送到B、A锁存器，要写入寄存器的数据由一个多路选择器送来。

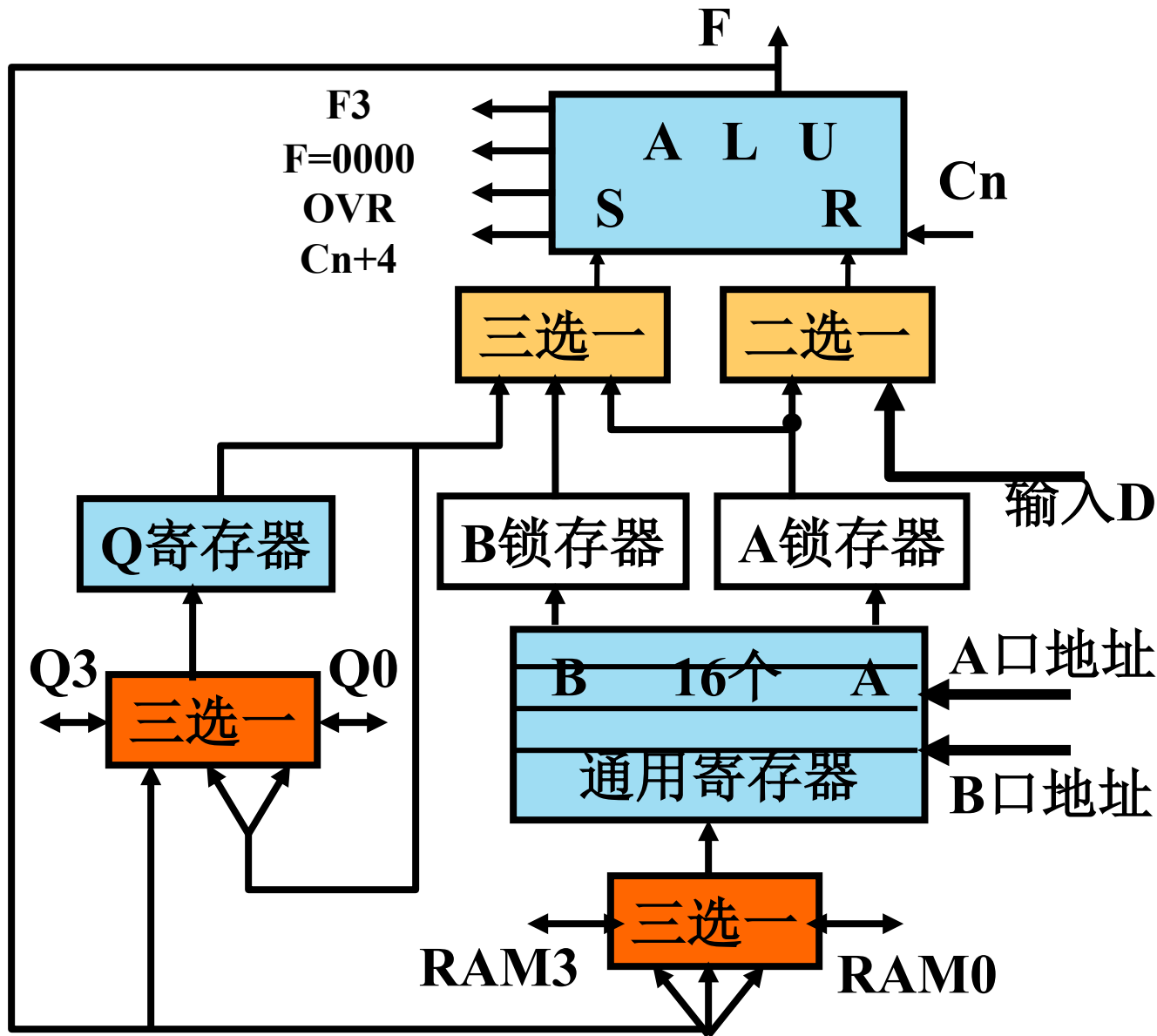


第三个组成部分是乘商寄存器Q，它能为自己的内容完成左右移位功能，其输出可以送往ALU，并可接收ALU的输出结果。



一组三选一门和另一组二选一门用来选择送向ALU的R、S输入端的数据来源，包括Q寄存器、A口、B口、外部输入D数据的8种不同组合。

第四个是5组门,包括



一组三选一门完成把ALU的输出、或左移一位、或右移一位的值送往通用寄存器组，最高、最低位移位信号有双向入/出问题

一组三选一门完成Q寄存器的左移一位、或右移一位、或接收ALU输出值的功能，最高、最低位移位信号有双向入/出问题。



输出Y

/OE

F

二选一

F3  
F=0000  
OVR  
Cn+4

A L U

S

R

Cn

三选一

二选一

输入D

Q寄存器

B锁存器

A锁存器

Q3

Q0

三选一

B

16个

A

A口地址

通用寄存器

B口地址

RAM3

三选一

RAM0

一组二选一门，选择把A口数据或ALU结果送出芯片，以给出输出Y的数据，Y输出的有无还受输出使能 /OE信号的控制，仅当 /OE为低是才有Y输出， /OE为高，Y输出为高阻态。

# 8 种结果处理

3位控制码	通用寄存器	Q寄存器	Y输出
000		$Q \leftarrow F$	F
001			F
010	$B \leftarrow F$		A
011	$B \leftarrow F$		F
100	$B \leftarrow F/2$	$Q \leftarrow Q/2$	F
101	$B \leftarrow F/2$		F
110	$B \leftarrow 2F$	$Q \leftarrow 2Q$	F
111	$B \leftarrow 2F$		F

# Am2901的控制信号

编码	<sup>B</sup> I8	<sup>Q</sup> I7 I6	<sup>Y</sup>	I5 I4 I3	<sup>R</sup> I2	<sup>S</sup> I1 I0
000		$Q \leftarrow F$	F	$R + S$	A	Q
001			F	$S - R$	A	B
010	$B \leftarrow F$		A	$R - S$	0	Q
011	$B \leftarrow F$		F	$R \vee S$	0	B
100	$B \leftarrow F/2$	$Q \leftarrow Q/2$	F	$R \wedge S$	0	A
101	$B \leftarrow F/2$		F	$\overline{R} \wedge S$	D	A
110	$B \leftarrow 2F$	$Q \leftarrow 2Q$	F	$R \oplus S$	D	Q
111	$B \leftarrow 2F$		F	$\overline{R \oplus S}$	D	0

输出Y

/OE

二选一

F

F3  
F=0000  
OVR  
Cn+4

A L U

S

R

Cn

三选一

二选一

输入D

Q寄存器

B锁存器

A锁存器

Q3

Q0

三选一

B 16个 A

通用寄存器

A口地址

B口地址

RAM3

三选一

RAM0

运算器，三大件  
运算 暂存 乘除快  
多路选通连起来

数据组合有内外  
运算功能指明白  
存移输出巧安排

运算功能选择

← I5 I4 I3

数据组合选择

← I2 I1 I0

运算结果处理

← I8 I7<sub>28</sub> I6

输出Y

/OE

F

二选一

Am2901内部组成

I5~I3



I2~I0



F3  
F=0000  
OVR  
Cn+4

A L U  
S R

Cn

三选一

二选一

输入D

乘商寄存器Q

B锁存器

A锁存器

Q3 Q0

三选一

A口地址

B 16个 A  
通用寄存器

B口地址

功能

8种运算功能

8种数据组合

8种结果处理

RAM3

三选一

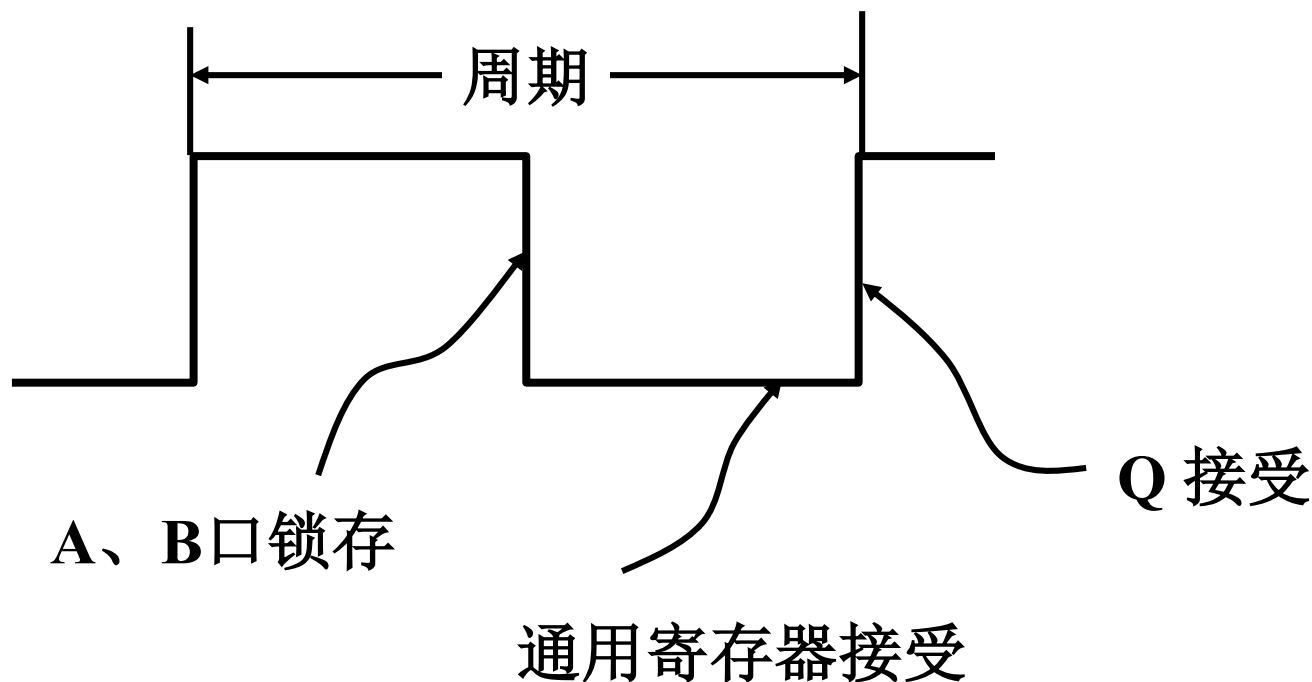
RAM0

I8~I6



# 运算器的时钟脉冲信号

## 运算器的时钟脉冲信号 CP

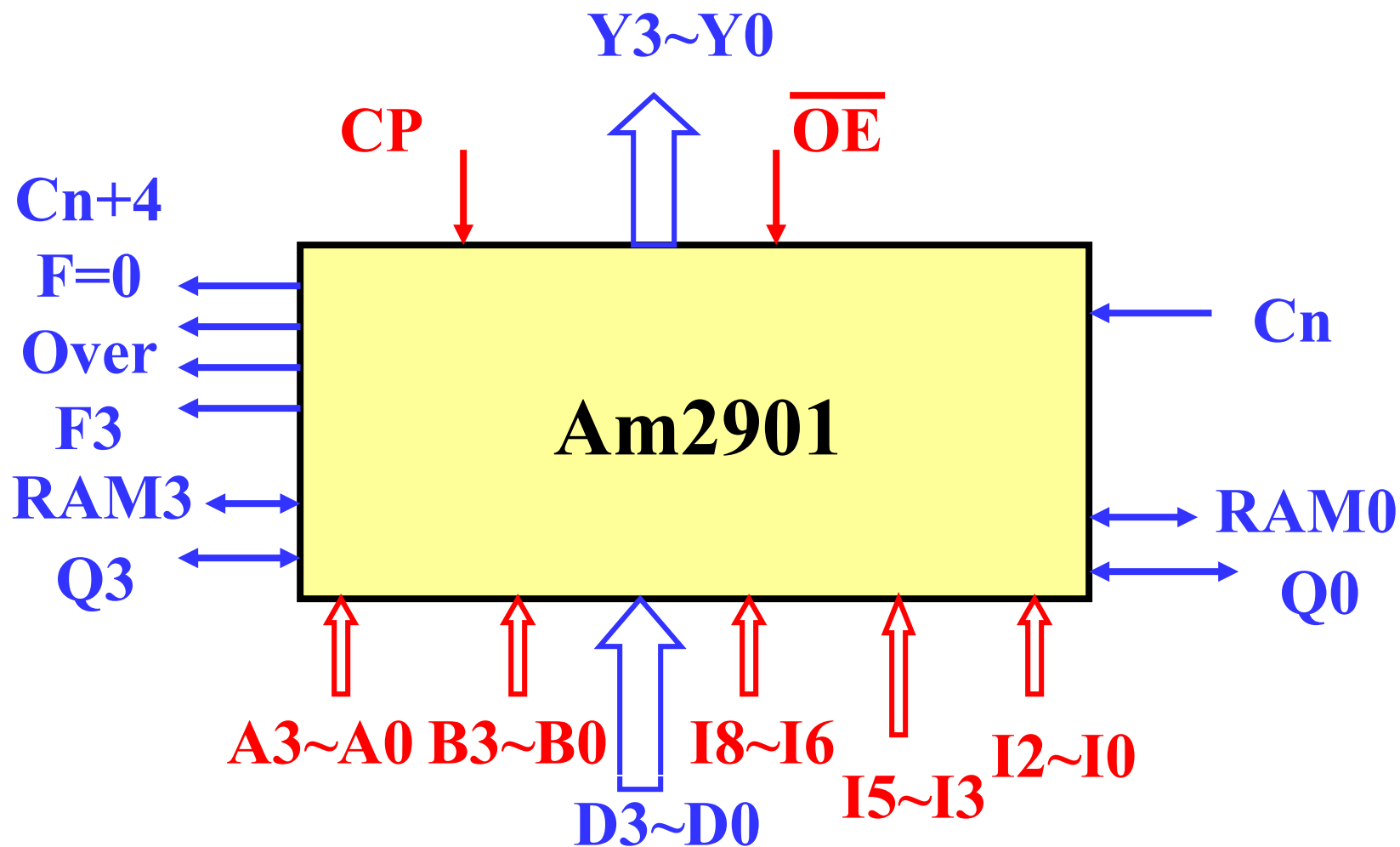


注意两个跳变沿  
和低电平的作用

# Am2901的操作使用

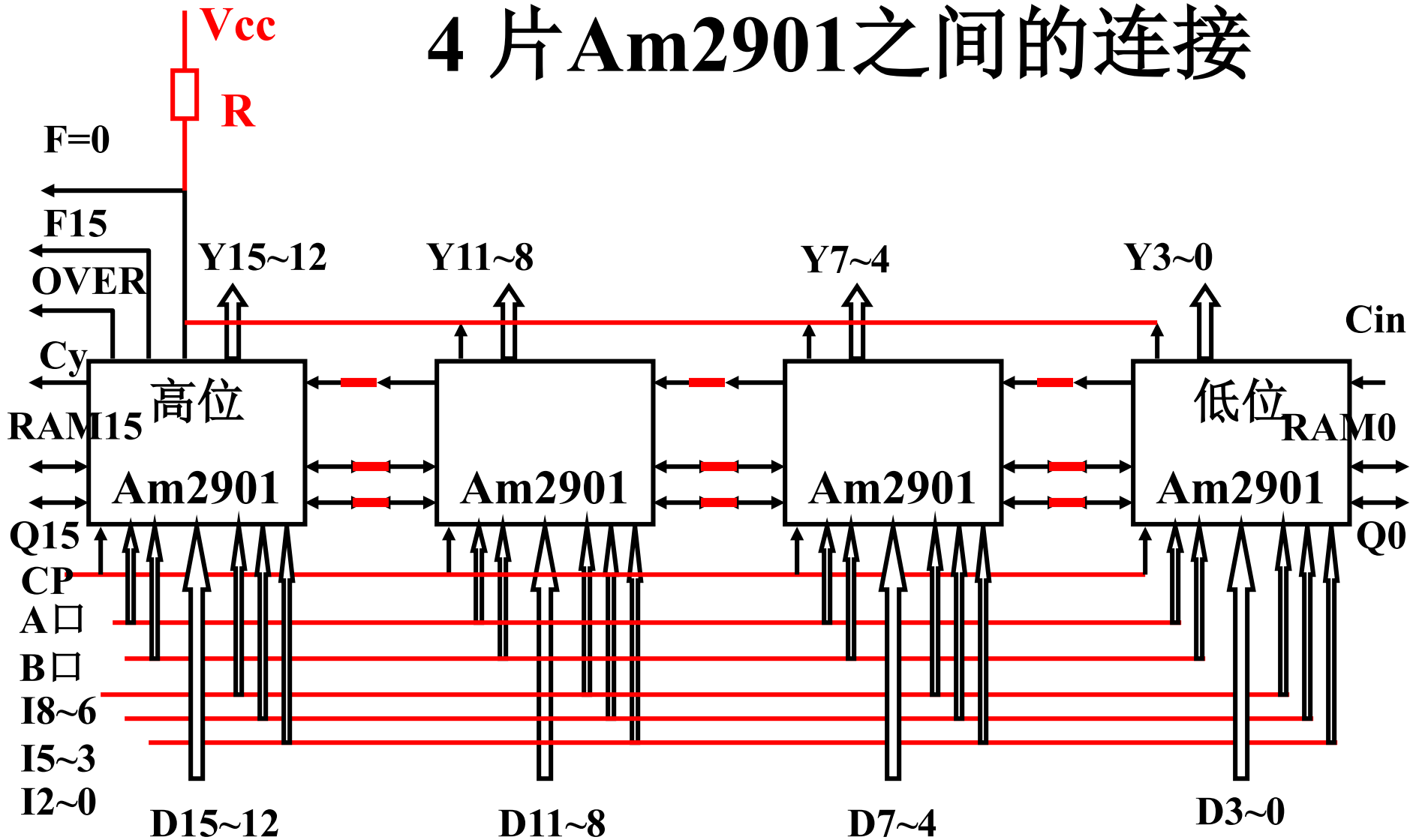
操作功能	控制信号					
	B口	A口	I8 I7 I6	I5 I4 I3	I2 I1 I0	Cn
$R0 \leftarrow R0 + R1$	0000	0001	011	000	001	0
$R2 \leftarrow R2 - R0$	0010	0000	010	001	001	1
右移 $R0 \leftarrow R0 + R1$	0000	0001	101	000	001	0
$Q \leftarrow R0 \quad Y \leftarrow F$	/	0000	000	000	100	0
$R0 \leftarrow R0 - R1 \quad Y \leftarrow A口$	0000	0001	010	100	001	0
$R0 \leftarrow R0 \vee R1 \quad Y \leftarrow F$	0000	0001	011	100	001	0

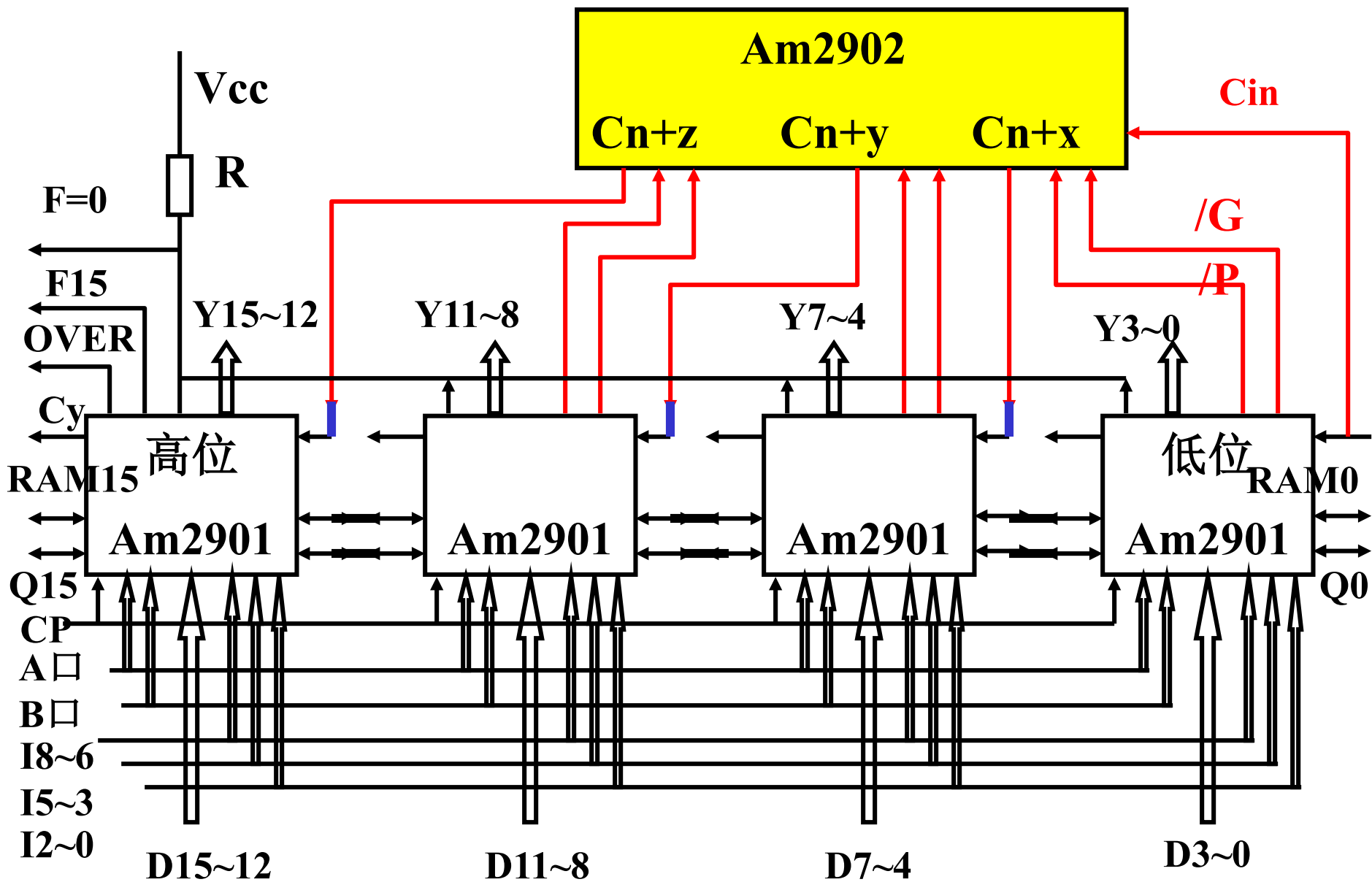
# 入出信号及引脚



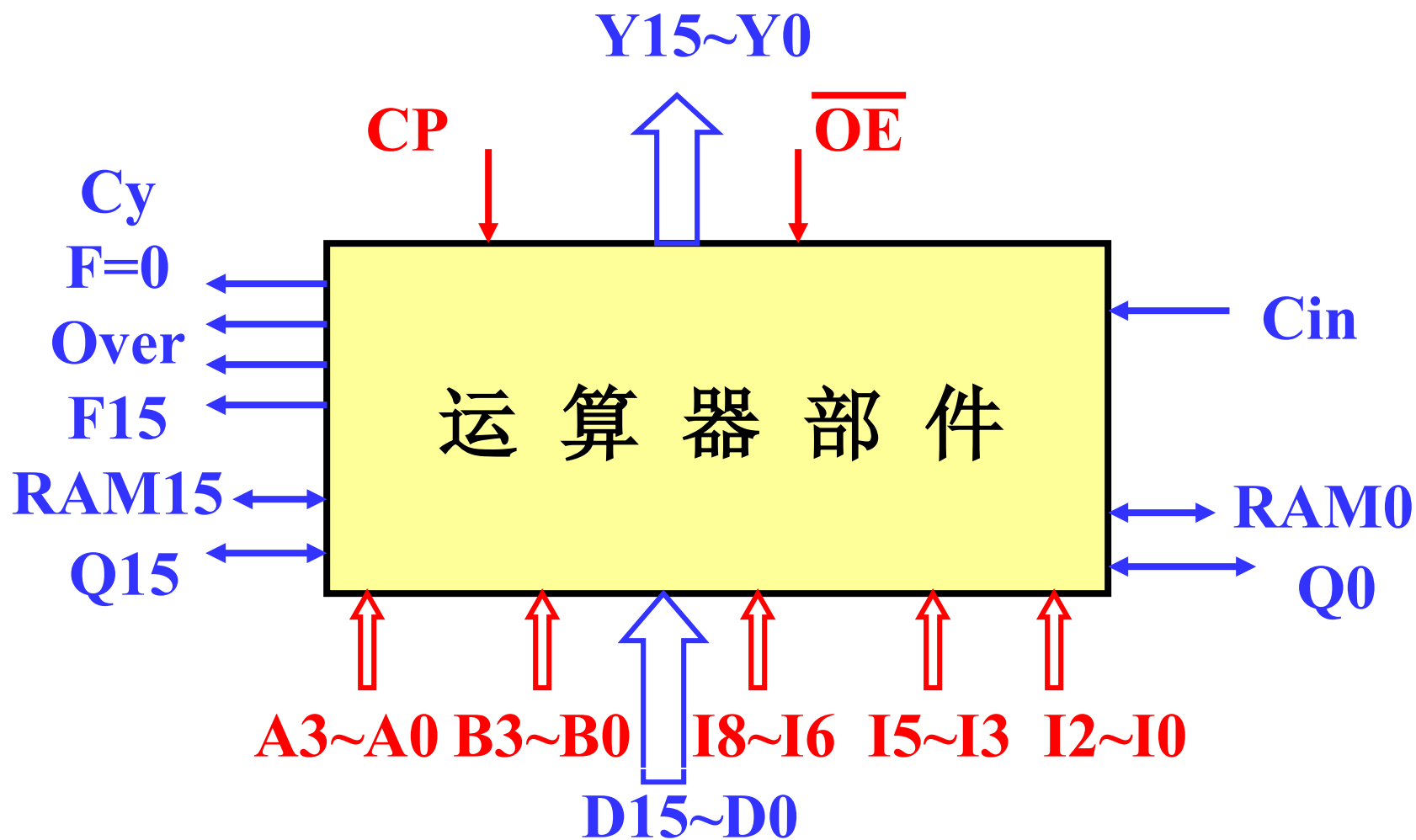


# 4 片 Am2901 之间的连接





# 入出信号及引脚



# 运算器实用中的几个问题

- 运算器最低位进位输入信号  $C_{in}$  的形成
- 运算器4位标志位信号的接收与记忆线路
- 移位时最高、最低位的移位输入信号设计

# 如何提供ALU最低位进位信号

**SCI**

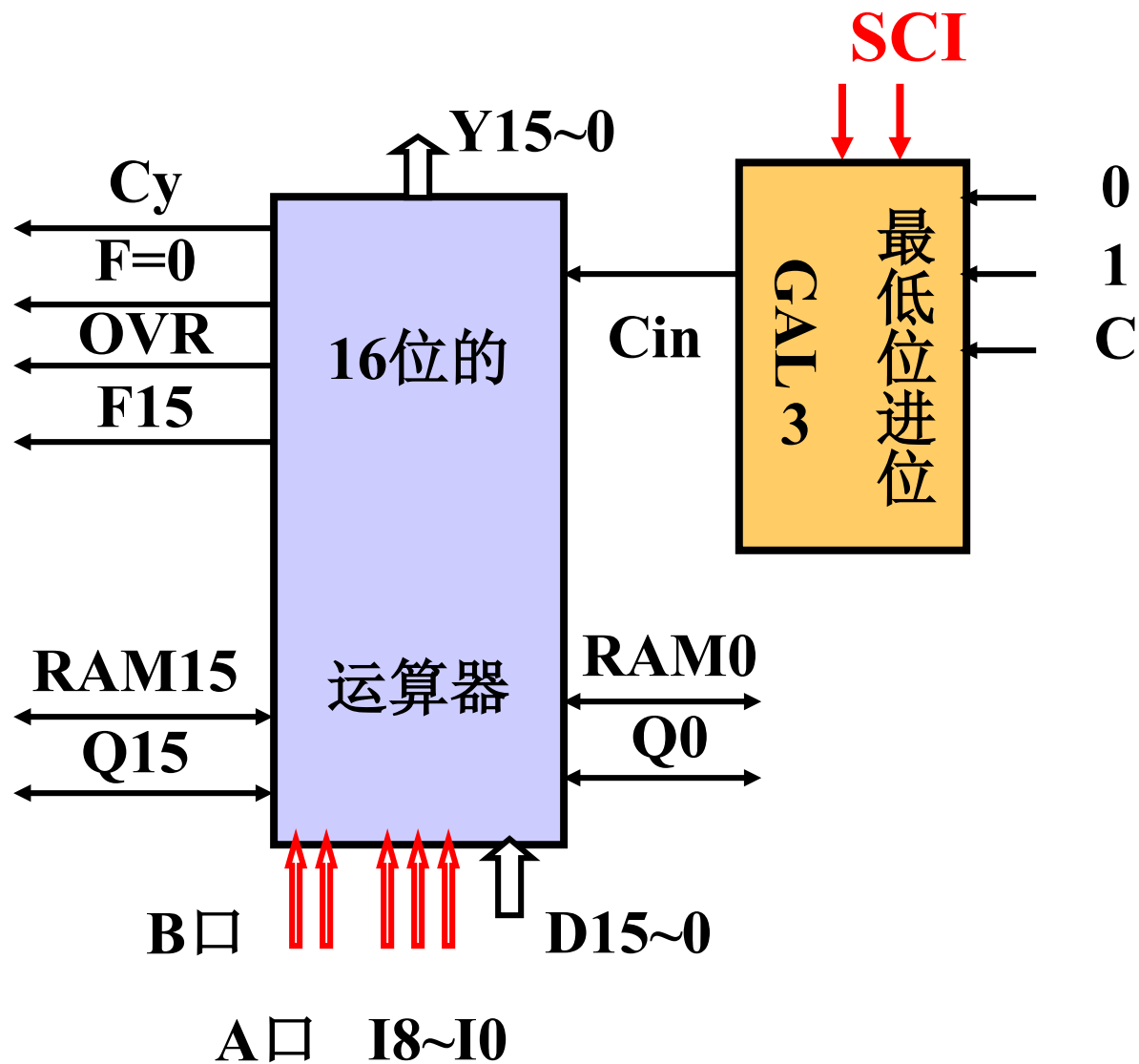
不同指令需求	Cin	例子最低位进位	控制码
ADD指令	0	$R0+R1 + 0$	00
INC指令	1	$R0+0 + 1$	01
ADC指令	C	$R0+R1 + C$	10
SUB指令	1	$R0+/-R1 + 1$	01
DEC指令	0	$R0+/-0 + 0$	00
SBB指令 (原理)	C	$R0+/-R1 + C$	10

最低位进位信号由 **SCI** 编码决定

# 最低位进位信号Cin的逻辑表达式

$$\begin{aligned} \mathbf{Cin} &= \neg \mathbf{SCI1} * \mathbf{SCI0} \\ &+ \mathbf{SCI1} * \neg \mathbf{SCI0} * \mathbf{C} \end{aligned}$$

# 16 位运算器最低位进位输入信号Cin



# 如何处理计算结果标志位

**算术与逻辑的运算指令**结果产生标志位的值

该标志位的值只在执行运算步骤接收输入

算术运算指令的结果会影响 C、Z、V、S

逻辑运算指令的结果只影响 Z、S

其他指令或其他执行步骤不接受标志位的值  
由此可见：

若设置标志位寄存器，

会有依据什么指令的什么时刻接收或不接收  
ALU 输出的标志位的值这样两种处理



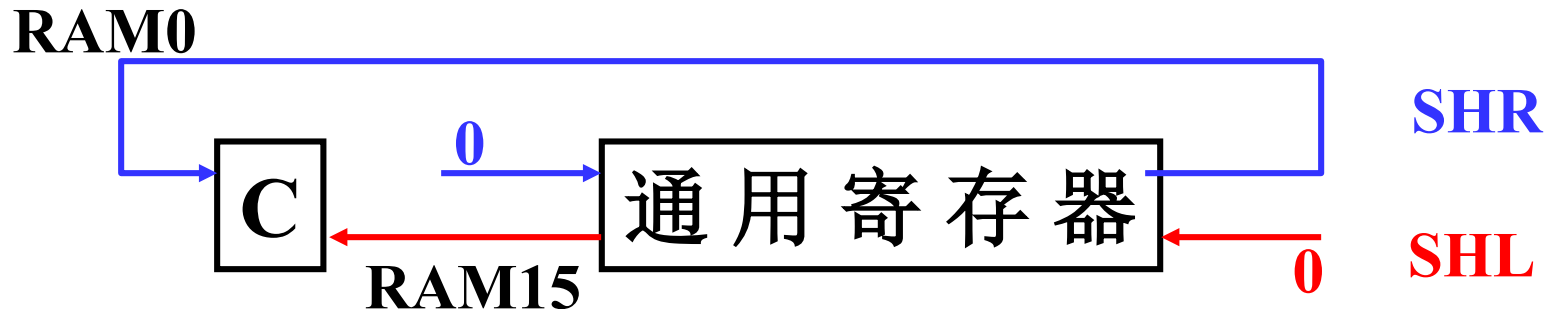
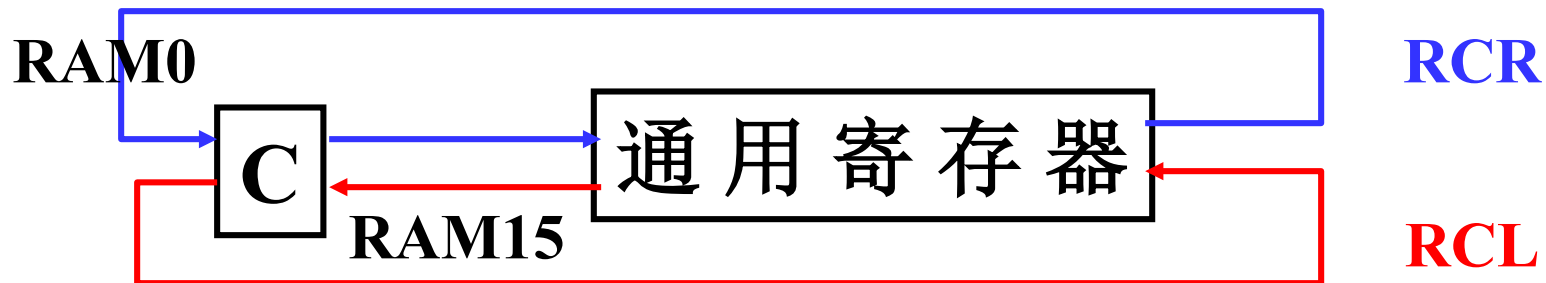
# 几条影响 C 触发器的指令

指令中有 STC ( $C \leftarrow 1$ )、CLC 指令 ( $C \leftarrow 0$ )

指令中有 RCL、RCR、SHL、SHR 指令，

它们的移位输出要传送进 C，

还需要决定寄存器的移位输入是什么值。

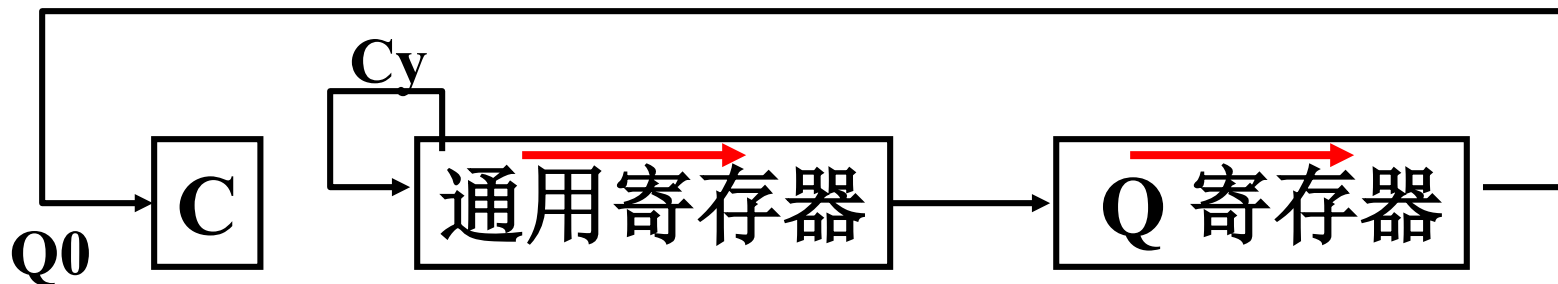


# 对乘法除法指令的支持

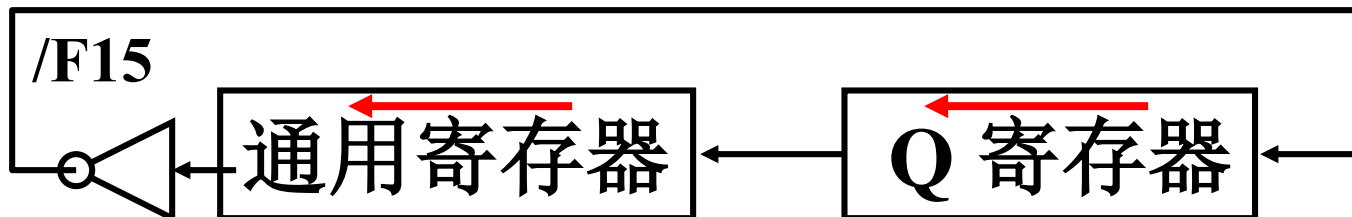
指令中 **MUL SR** 指令，**DIV SR** 指令

它们的运行用到了移位输出要传送进 **C**

(还决定寄存器的移位输入) 乘法，联合右移位



除法，联合左移位(即为商)



# 状态寄存器的控制

用3位码控制      有8处来源

SST 2~0	C	Z	V	S
000	不变	三位不变		(C,Z,V,S)
001	接受ALU状态输出 (Cy,ZR,OV,F15)			
010	接收内部总线输出 (IB7 6 5 4)			
011	0	三位不变		
100	1	三位不变		
101	RAM0	三位不变		
110	RAM15	三位不变		
111	Q0	三位不变		

# 状态寄存器的逻辑表达式

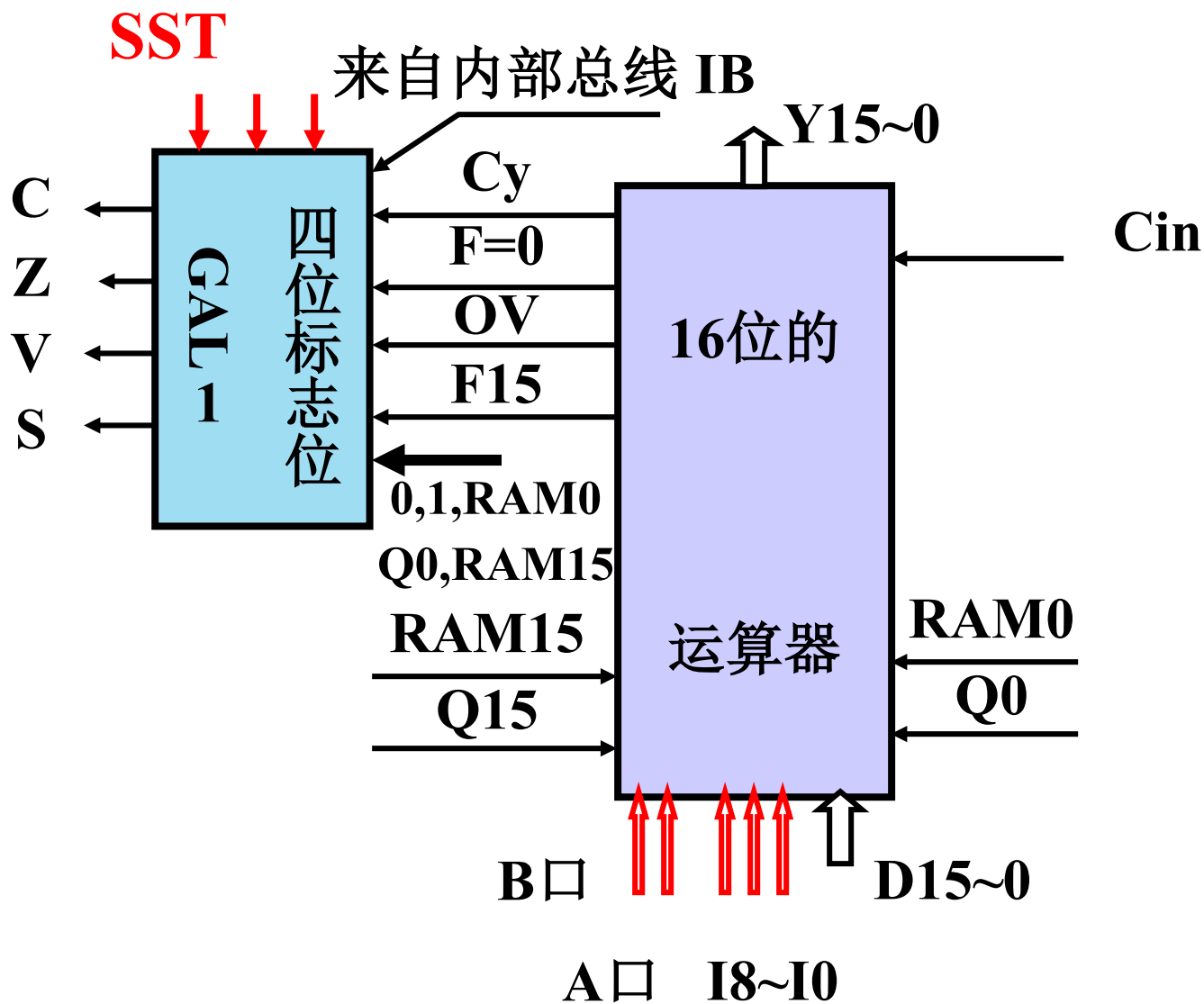
$$\begin{aligned} \mathbf{C} := & /SST2*/SST1*/SST0* C \\ & + /SST2*/SST1* SST0* Cy \\ & + /SST2* SST1*/SST0* IB7 \\ & + SST2*/SST1*/SST0 \\ & + SST2*/SST1* SST0* RAM0 \\ & + SST2* SST1*/SST0* RAM15 \\ & + SST2* SST1* SST0* Q0 \end{aligned}$$

$$\begin{aligned} \mathbf{Z} := & /SST2*/SST1*/SST0* Z \\ & + /SST2*/SST1* SST0* ZR \\ & + /SST2* SST1*/SST0* IB6 \\ & + /SST2* SST1* SST0* Z \\ & + SST2* Z \end{aligned}$$

$$\begin{aligned} \mathbf{V} := & /SST2*/SST1*/SST0* V \\ & + /SST2*/SST1* SST0* OV \\ & + /SST2* SST1*/SST0* IB5 \\ & + /SST2* SST1* SST0* V \\ & + SST2* V \end{aligned}$$

$$\begin{aligned} \mathbf{S} := & /SST2*/SST1*/SST0* S \\ & + /SST2*/SST1* SST0* F15 \\ & + /SST2* SST1*/SST0* IB4 \\ & + /SST2* SST1* SST0* S \\ & + SST2* S \end{aligned}$$

# 16 位运算器的状态寄存器



# 寄存器移位输入信号

2 位控制码 SSH1~0	左移(I7) RAM0 Q0	右移(/I7) RAM15 Q15	指令
00	0 /	0 /	SHL SHR
01	C /	C /	RCL RCR
10	Q15 /F15	Cy F0	DIV MUL

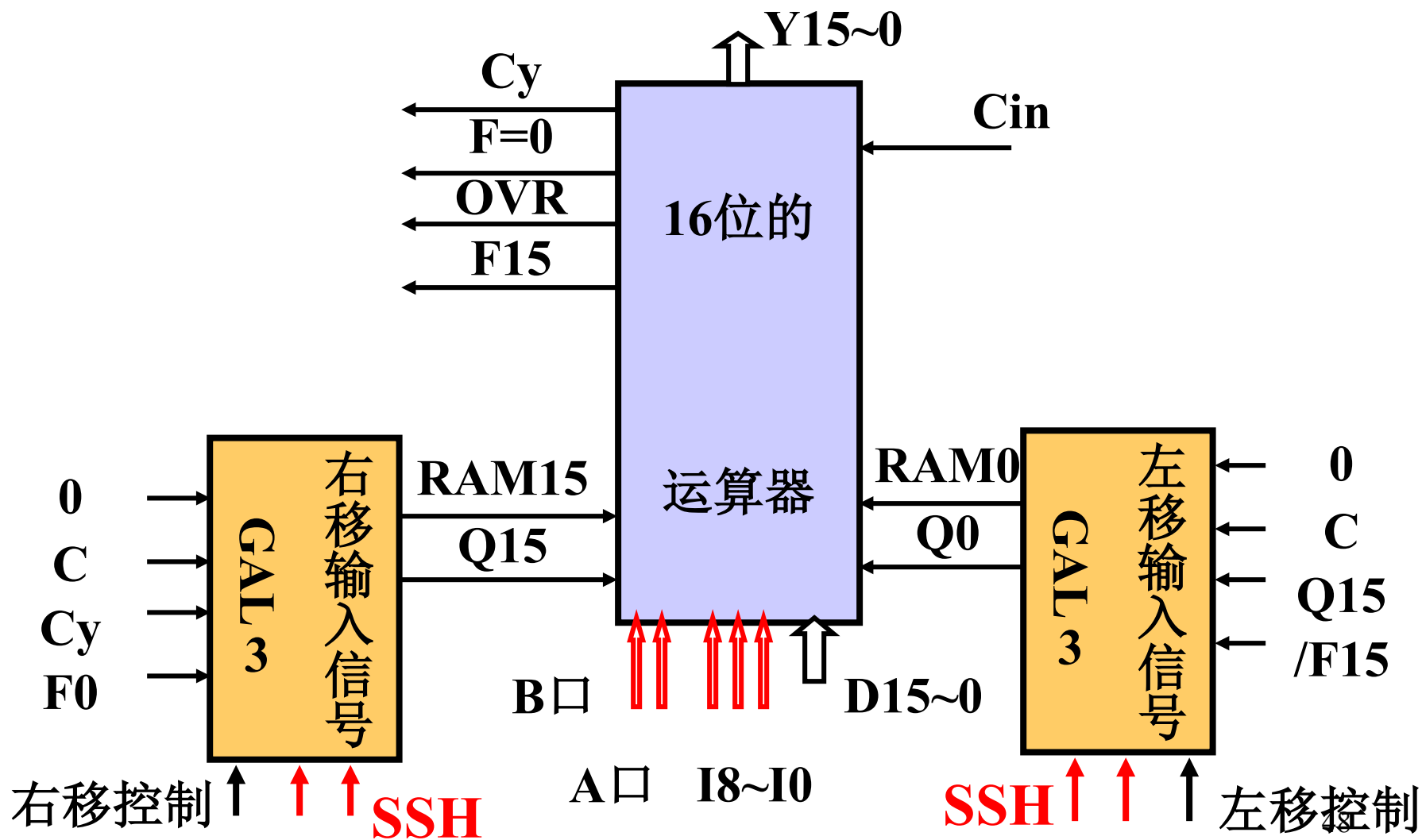
实现中，具体移位输入信号是用SSH 编码和 ALU 中的结果处理控制信号 I7（移位方向）共同决定的。

# 左右移位输入信号逻辑表达式

$$\begin{aligned}\text{RAM15} &= \text{/SSH1} * \text{SSH0} * \text{C} \\ &+ \text{SSH1} * \text{/SSH0} * \text{Cy} \\ \text{Q15} &= \text{SSH1} * \text{/SSH0} * \text{F0} \\ \text{RAM0} &= \text{/SSH1} * \text{SSH0} * \text{C} \\ &+ \text{SSH1} * \text{/SSH0} * \text{Q15} \\ \text{Q0} &= \text{SSH1} * \text{/SSH0} * \text{/F15}\end{aligned}$$

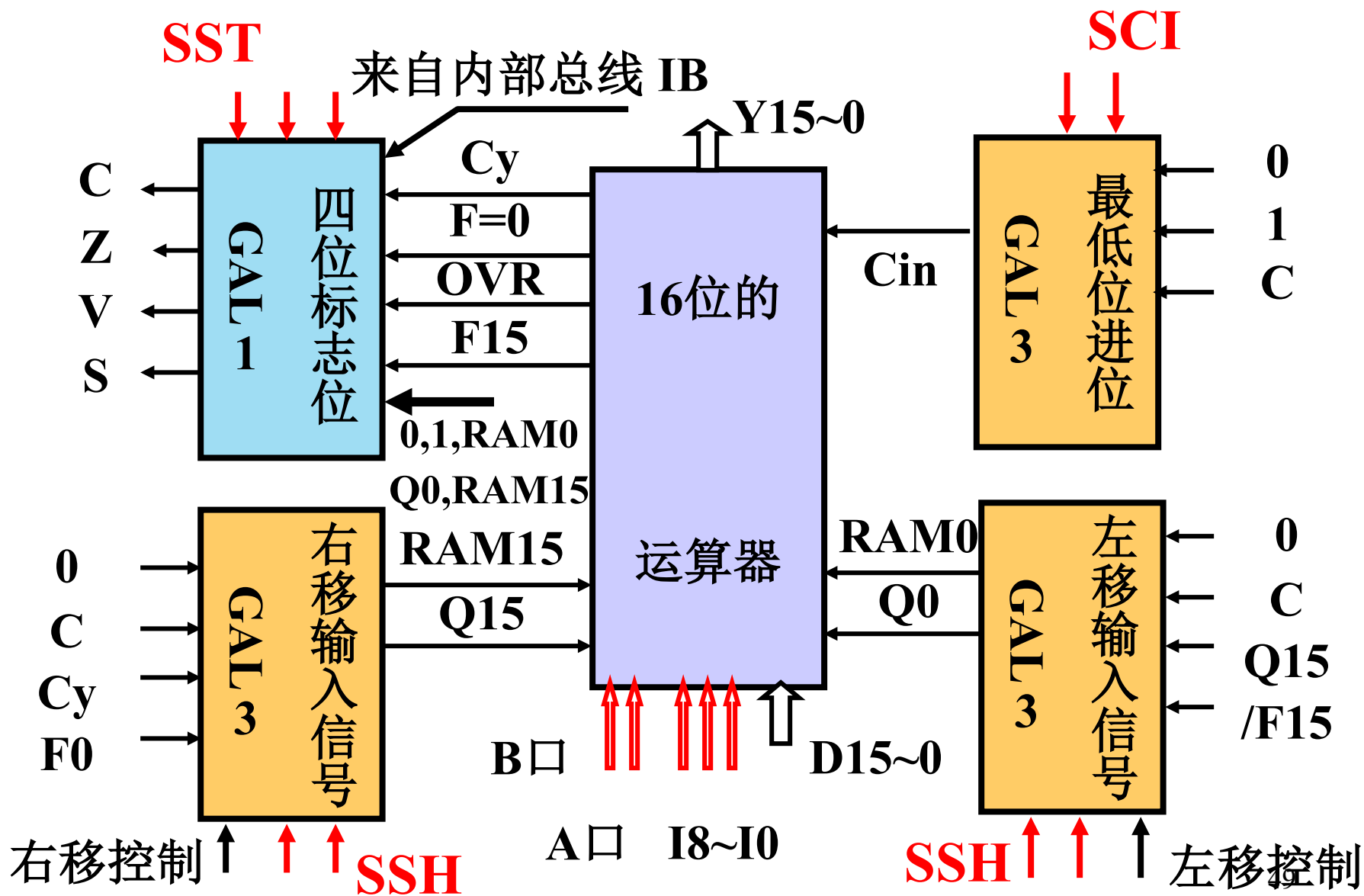
$$\begin{aligned}\text{RAM15} \text{ .OE} &= \text{/I7} \\ \text{Q15} \text{ .OE} &= \text{/I7} \\ \text{RAM0} \text{ .OE} &= \text{I7} \\ \text{Q0} \text{ .OE} &= \text{I7}\end{aligned}$$

# 16 位运算器的最高位、最低位移位输入信号





# 16 位运算器的完整组成



# 运算器用的GAL20V8的控制信号

**GAL1: 状态寄存器**

SST	C	Z	V	S
000	C	Z	V	S
001	Cy	ZR	OV	F15
010	内部总线的一位			
011	1	Z	V	S
100	0	Z	V	S
101	RAM0	Z	V	S
110	RAM15	Z	V	S
111	Q0	Z	V	S

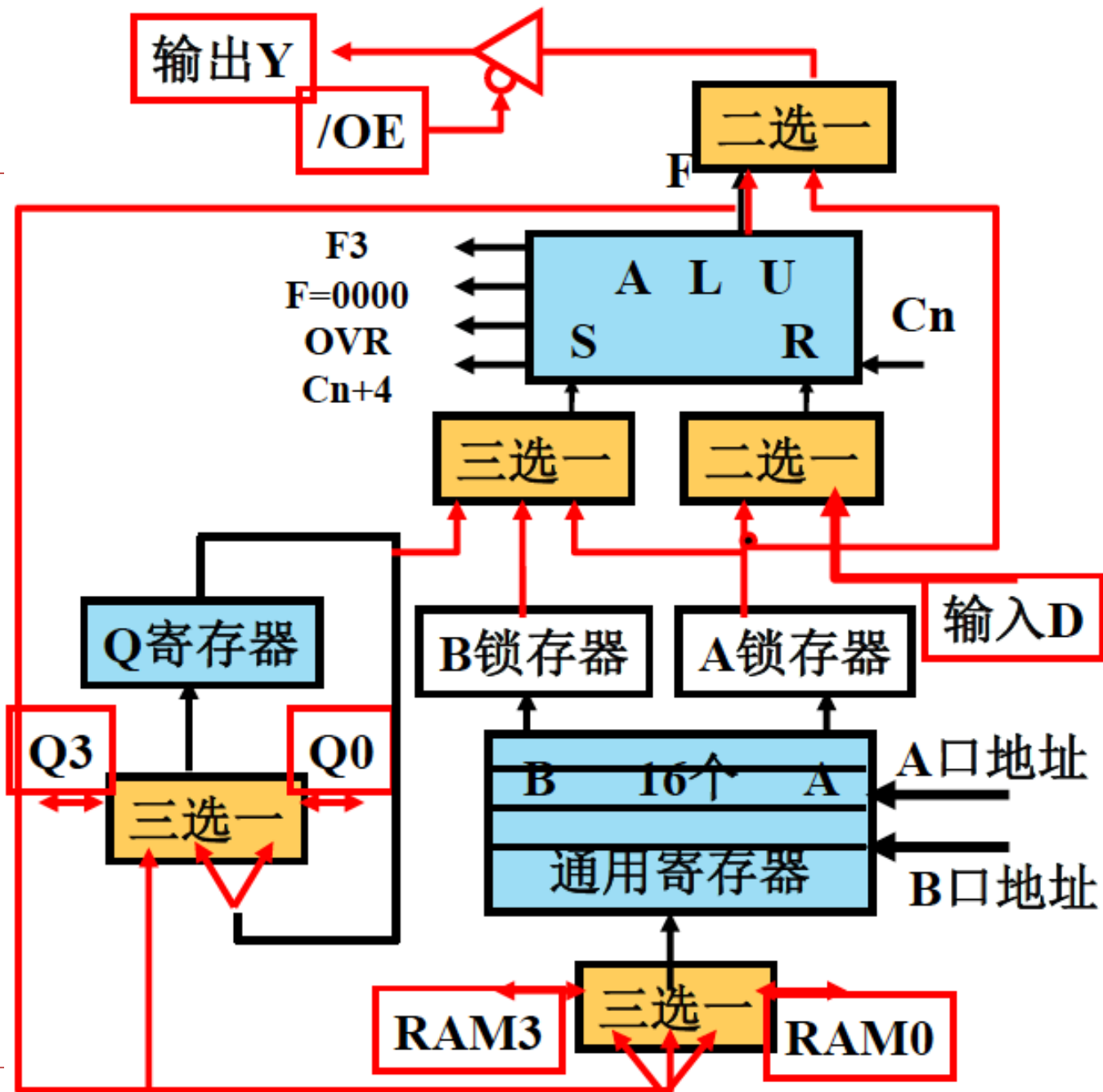
**GAL3:**                      进位输入

SCI	运算	Cin
00	加	0
01	加	1
10	加	C

移位输入

	RAM0 Q0	RAM15 Q15
SSH	左移位	右移位
00	0 X	0 X
01	C X	C X
10	Q15 /F15	Cy F0

# AM2901



# 小结

---

- ▶ 运算器基本组成
- ▶ 运算器的基本设计过程
- ▶ 运算器Am2901
  - ▶ 4位运算器
  - ▶ 8种运算功能
  - ▶ 8种数据来源组合
  - ▶ 8种数据输出方式
  - ▶ Verilog实现举例

# 阅读和思考

---

- ▶ 阅读

- ▶ Am2901的有关材料

- ▶ 思考

- ▶ 用Verilog语言实现完整的Am2901.

---

# 谢谢

