

计算机组成与系统结构

第二章 运算方法和运算器

吕昕晨

lvxinchen@bupt.edu.cn

网络空间安全学院

2020-6-23

第一章答疑

1.2.4 计算机的性能指标

例1.1: 对于一个给定的程序, I_N 表示执行程序中的指令总数, t_{CPU} 表示执行该程序所需的CPU时间, T 为时钟周期, f 为时钟频率 (T 的倒数), N_C 为CPU时钟周期数。设CPI表示每条指令的平均时钟周期数, MIPS表示每秒钟执行的百万条指令数, 请写出如下四种参数的表达式:

(1) t_{CPU} (2) CPI (3) MIPS (4) N_C

2020-6-23

第一章答疑

- 总时间: t_{CPU}
- 总周期: N_C
- CPU: 频率 f / 周期 T (互为倒数)
- 指令: 总数 I_N / 速率MIPS
- 指令 v.s. CPU: CPI (每条指令时钟周期数)

- 总时间: t_{CPU}
 - CPU: $T \cdot (1/f) \cdot N_C \cdot (I_N \cdot CPI)$ (第一问)
 - MIPS: $I_N / (MIPS \cdot 10^6)$ (第三问)
- 转换参数: CPI
 - $N_C \cdot (\sum CPI_i \cdot I_i) / I_N$ (第二/四问)
 - MIPS: $f / (MIPS \cdot 10^6)$ (MIPS) (第三问)

第一章答疑

解:

$$(1) t_{CPU} = N_C \times T = N_C \times f \times T = I_N \times CPI \times T = (\sum_{i=1}^n CPI_i \times I_i) \times T$$

$$(2) CPI = \frac{N_C}{I_N} = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_N} \quad \frac{I_i}{I_N} \text{ 表示指令在程序中所占比例}$$

$$(3) MIPS = \frac{I_N}{t_{CPU} \times 10^6} = \frac{f}{CPI \times 10^6}$$

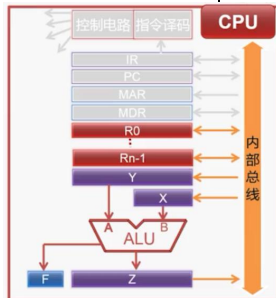
$$(4) N_C = \sum_{i=1}^n (CPI_i \times I_i)$$

上式中, I_i 表示 i 指令在程序中执行的次数, CPI_i 表示 i 指令所需的平均时钟周期数, n 为指令种类。

2020-6-23

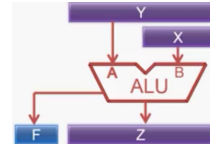
模型机——CPU 运算器

- 运算器用于进行算数运算和逻辑运算
- 算数运算
 - 加、减、乘、除
- 逻辑运算
 - 非、与、或
- 数表示方式
 - 定点数
 - 浮点数



本周教学安排

- 直播内容
 - 逻辑运算
 - 非、与、或
 - 算数运算
 - 加、减法
- 录播内容
 - 数的表示方法
 - 定点、浮点数、原、反、补码
 - 字符串与汉字等
 - 逻辑运算



第二章 运算方法和运算器

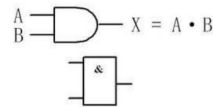
- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算
- 超前进位加法器
- 算数逻辑单元实现

2020-6-23

7

各类逻辑运算与逻辑门（1）

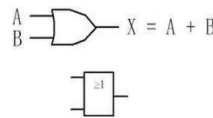
- 与门



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

与操作真值表

- 或门

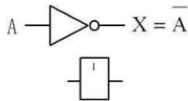


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

或操作真值表

各类逻辑运算与逻辑门（2）

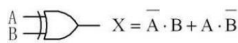
- 非门



A	X
0	1
1	0

非操作真值表

- 异或门



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

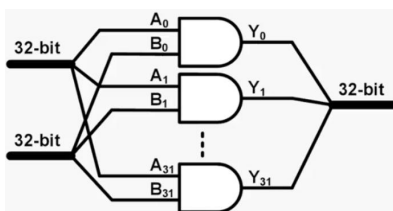
异或操作真值表

运算器中逻辑运算构建

- 逻辑门——1位二进制数
- 运算器逻辑运算指令
 - 与：AND A, B
 - 或：OR A, B
 - 非：NOT A
 - 异或：XOR A, B
 - 上述A/B寄存器为多位（例如，80386为32位）
- 问题
 - 1) 将1位逻辑门扩展为32位逻辑运算电路
 - 2) 如何实现多种逻辑运算

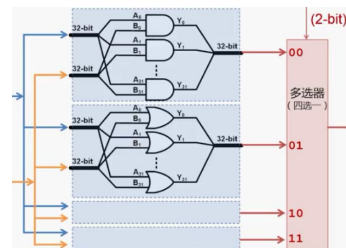
问题1：与运算的位扩展

- 逻辑运算特性——按位运算
- 位扩展方式：32个与门分别连接



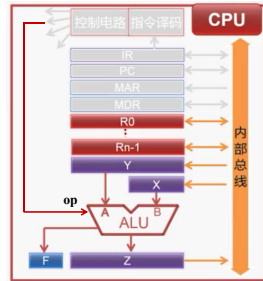
问题2：多种逻辑运算

- 逻辑运算单元：包含多种逻辑功能
- 扩展方式：多种逻辑功能单元+多路器+选通信号



逻辑运算单元

- 逻辑运算单元→ALU
 - 选通控制信号
 - 两位二进制: op
- | op1 | op2 | 功能 |
|-----|-----|----|
| 0 | 0 | 与 |
| 0 | 1 | 或 |
| 1 | 0 | 非 |
| 1 | 1 | 异或 |
- 在ALU新增op信号输入端
 - 控制ALU逻辑运算功能



单选题 1分

此题未设置答案，请点击右侧设置按钮

已知 A=10010010, B=00101111
1) AND A, B
2) XOR A, B

- ☐ A 1) 00000010; 2) 10111101
- ☐ B 1) 01000010; 2) 10100101
- ☐ C 1) 00001110; 2) 10111001
- ☐ D 1) 00010010; 2) 10100101

2020-6-23

提交

14

填空题 2分

此题未设置答案，请点击右侧设置按钮

逻辑门可实现1位二进制单一功能逻辑运算；

为了实现ALU的多位多功能逻辑运算功能，需要进行[填空1]和[填空2]。

- 1) 位扩展；
- 2) 功能扩展

正常使用填空题需3.0以上版本浏览器

2020-6-23

作答

15

第二章 运算方法和运算器

- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算
- 超前进位加法器
- 算数逻辑单元实现

2020-6-23

16

二进制加法示例

- 4位二进制加法——A+B
- 示例: A=1101, B=0101

$$\begin{array}{r}
 A \quad 1101 \\
 + B \quad 0101 \\
 \hline
 A+B \quad 10010
 \end{array}$$

- 回顾逻辑运算设计思路
 - 问题1: 设计1位二进制加法器 (本节)
 - 问题2: 设计多位加法器 (行波进位、超前进位)
 - 问题3: 实现多种算数功能 (加/减法)

17

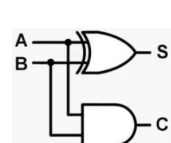
半加器

- 半加器功能: 两个1位二进制数相加
 - 输入端口: A、B
 - 输出端口: S (和)、C (进位)

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

异或

与



18

全加器

- 全加器功能：低位进位
- 三个1位二进制数相加
 - 输入：A、B、 C_{in} （进位输入）
 - 输出：S、 C_{out} （进位输出）

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = A_i B_i + A_i C_i + B_i C_i$$

$$= A_i B_i + (A_i \oplus B_i) C_i$$

逻辑函数

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

19

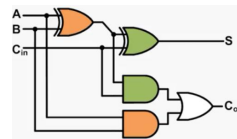
全加器实现

$$S_i = A_i \oplus B_i \oplus C_i$$

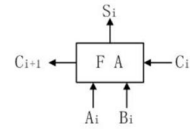
$$C_{i+1} = A_i B_i + A_i C_i + B_i C_i$$

$$= A_i B_i + (A_i \oplus B_i) C_i$$

(a) 逻辑函数



(b) 全加器门电路



(c) 全加器框图

20

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

全加器逻辑函数可表示为：

☐ A

$$S = AB + BC_{in}$$

$$C_{out} = AB + (A \oplus B)C_{in}$$

☐ B

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + BC_{in} + AC_{in}$$

2020-6-23

提交

21

第二章 运算方法和运算器

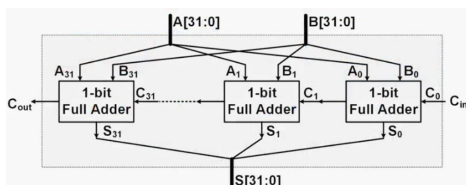
- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算
- 超前进位加法器
- 算数逻辑单元实现

2020-6-23

22

问题2：设计多位加法器

- 扩展方式：全加器级联扩展
- 将前一个全加器的 C_{out} 连接到后一个全加器 C_{in}



23

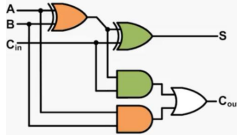
行波进位加法器分析

- 结构特点
 - 级联：前一个全加器进位输出 → 后一个全加器进位输入
- 优点
 - 电路布局简单，设计方便
- 缺点
 - 后一级全加器输入需等待前一级全加器输出
 - 延迟时间长，且随加法位数递增
- 如何分析行波进位加法器时延？

24

延迟分析 (1)

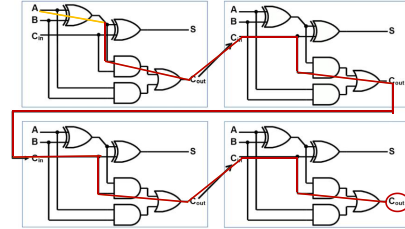
- 基本门电路延迟
 - 与门/或门延迟: T
 - 异或门延迟: $3T$
- 每级全加器
 - 输入A、B → 输出S: $6T$
 - 输入Cin → 输出Cout: $2T$



25

延迟分析 (2)

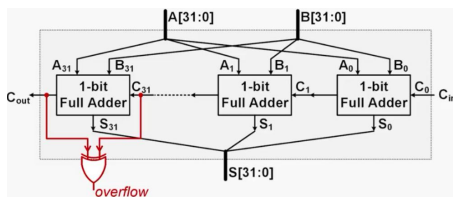
- n位Cout延迟: $t_a = 3T + 2T * n$
 - 第一级异或门: $3T$
 - 每级进位输入输出: $2T$



26

行波进位加法器

- 溢出判断: **最高位进位输入不等于其进位输出**
 - 有符号数 v.s. 无符号数
 - 溢出判断分析 (后续课件)
 - 溢出判断延迟: $t_a = 3T + 2T * n + 3T$



27

单选题 1分

此题未设置答案, 请点击右侧设置按钮

N位行波进位加法器中, 最高位进位输出 Cout 与溢出判断延迟分别为?

- ☐ A
- $2T * N + 3T$
 $2T * N + 4T$
- ☒ B
- $2T * N + 3T$
 $2T * N + 6T$

2020-6-23

提交

28

第二章 运算方法和运算器

- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算**
- 超前进位加法器
- 算数逻辑单元实现

2020-6-23

29

有符号数存储方式——补码

- 减法运算可转换为加法运算

$$A - B = A + (-B)$$
- 算数 (负号) \longleftrightarrow 负号: 表示补码
- 转换规则:
 - 负号: 按位取反, 末位加一
- $[-1]$ 的原/反/补码

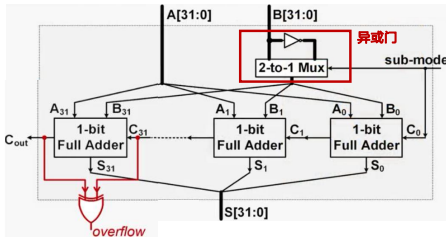
$$= [10000001]_{\text{原}} = [11111110]_{\text{反}} = [11111111]_{\text{补}}$$
- 加减法转换

$$A - B = A + (B)_{\text{补}}$$

30

加减法运算实现

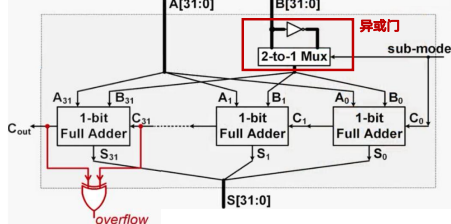
- 实现思路：加法器基础上进行修正
 - B输入端加入多路器
 - sub-mode 选择信号：0 / 1



31

延迟分析

- 不带减法功能时延： $t_a = 6T + 2T * n$
- 判断加减法运算（第一级异或门）： $3T$
- 加减法器时延： $t_a = 9T + 2T * n$ → 避免行波进位加法器延迟？

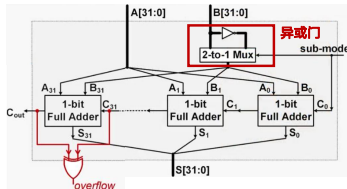
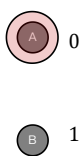


32

单选题 1分

此题未设置答案，请点击右侧设置按钮

下图所示电路中，若要实现加法运算，此时，sub-mode信号应为



2020-6-23

提交

33

第二章 运算方法和运算器

- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算
- 超前进位加法器
- 算数逻辑单元实现

2020-6-23

34

行波进位加法器优化

- 主要问题
 - 高位运算需等待低位运算结果
- 如何避免行波进位加法器延迟？
- 优化方法
 - 提前计算进位信号
 - 超前（先行）进位加法器
- 超前进位信号发生器设计
 - 分析进位逻辑表达式
 - 表达式化简 → 计算超前进位表达式

35

进位信号分析

$$C_{i+1} = A_i \cdot B_i + A_i \cdot C_i + B_i \cdot C_i \\ = A_i \cdot B_i + (A_i + B_i) C_i$$

- 超前进位
 - 避免 C_i 与 C_{i+1} 间依赖
- 令
 - 生成信号： $G_i = A_i \cdot B_i$
 - 传播信号： $P_i = A_i + B_i$
- 简化进位信号为

$$C_{i+1} = G_i + P_i \cdot C_i$$

36

进位信号计算

$$\begin{aligned}
 C_1 &= G_0 + P_0 \cdot C_0 \\
 C_2 &= G_1 + P_1 \cdot C_1 \\
 &= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0) \\
 &= G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0 \\
 C_3 &= G_2 + P_2 \cdot C_2 \\
 &= G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0) \\
 &= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0 \\
 C_4 &= G_3 + P_3 \cdot C_3 \\
 &= G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0) \\
 &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0 \\
 C_5 &= G_4 + P_4 \cdot G_3 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 \\
 &\quad + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0
 \end{aligned}$$

37

单选题 1分

此题未设置答案，请点击右侧设置按钮

设置

超前进位信号发生器中，进位信号 C_5 表达式为

A $G_4 + P_4 \cdot G_3 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0$

B $G_4 + P_4 \cdot G_3 + P_4 \cdot P_3 \cdot G_2 + P_4 \cdot P_3 \cdot P_2 \cdot G_1 + P_4 \cdot P_3 \cdot P_2 \cdot P_1 \cdot G_0$

2020-6-23

提交

38

进位信号分析（1）

- 延迟分析：3T（与级数无关）
 - 生成信号/传播信号延迟：T（与/或门）
 - 进位信号：2T（与门+或门）

$$\begin{aligned}
 C_1 &= G_0 + P_0 \cdot C_0 \\
 C_2 &= G_1 + P_1 \cdot C_1 \\
 &= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0) \\
 &= G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0 \\
 C_3 &= G_2 + P_2 \cdot C_2 \\
 &= G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0) \\
 &= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0 \\
 C_4 &= G_3 + P_3 \cdot C_3 \\
 &= G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0) \\
 &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0
 \end{aligned}$$

39

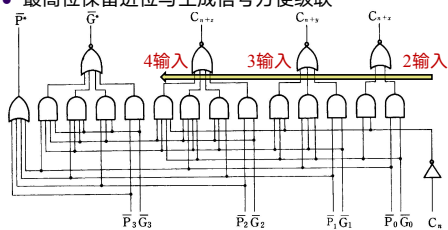
进位信号分析（2）

$$\begin{aligned}
 C_1 &= G_0 + P_0 \cdot C_0 \quad \text{2项} \\
 C_2 &= G_1 + P_1 \cdot C_1 \\
 &= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_0) \\
 &= G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0 \quad \text{3项} \\
 C_3 &= G_2 + P_2 \cdot C_2 \\
 &= G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_0) \\
 &= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0 \quad \text{4项} \\
 C_4 &= G_3 + P_3 \cdot C_3 \\
 &= G_3 + P_3 \cdot (G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_0) \\
 &= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot C_0 \quad \text{5项}
 \end{aligned}$$

40

超前进位信号发生器实现

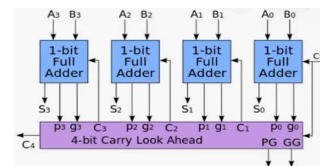
- 超前进位信号发生器实现（74182）：
 - 对照进位信号分析结果
- 最高位保留进位与生成信号方便级联



41

超前进位加法器实现

- 超前进位加法器
 - Carry-Lookahead Adder, CLA
- 实现方法
 - 并联加法器
 - 超前进位信号发生器



42

超前进位加法器分析

- 结构特点
 - 并联方法：统一进位信号发生器
- 优点：
 - 后一级输入不依赖前一级输出，电路延迟低
- 缺点：
 - 进位信号发生器实现复杂，并随位数增加，难以实现
- 实际加法器实现
 - 混合方式：行波进位（片间）+超前进位（片内）
 - 32位加法器：4个8位超前进位加法器

43

单选题 1分

此题未设置答案，请点击右侧设置按钮

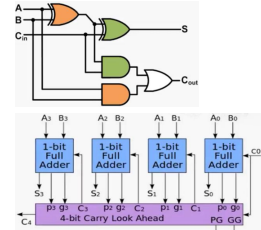
设置

已知：超前信号发生器时延为 $3T$ 。
全加器与超前进位加法器结构如下图所示。
N位超前进位加法器的最高位进位输出延迟为：

- A $2T * N + 3T$
 B $8T$
 C $5T$

2020-6-23

提交



第二章 运算方法和运算器

- 逻辑运算器
- 定点加法运算
- 行波进位加法器
- 定点减法运算
- 超前进位加法器
- 多功能算数逻辑单元

2020-6-23

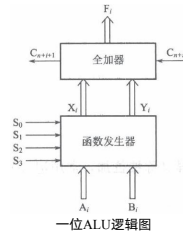
45

多功能算数逻辑单元 (ALU)

- 目标：
 - 实现多功能算数逻辑运算
 - 算数：加、减
 - 逻辑运算：与、或、非、异或等
- 方法：
 - 新增控制输入端：S0~S4
 - 配合函数发生器，实现不同功能
- 全加器 (FA) 输入为函数发生器输出

$$F_i = X_i \oplus Y_i \oplus C_{n+1}$$

$$C_{n+i+1} = X_i Y_i + Y_i C_{n+1} + C_{n+1} X_i$$



46

函数发生器功能

- 输入输出对应关系
 - S0~S1控制 Y_i
 - S2~S3控制 X_i

函数关系表

S_0	S_1	Y_i	S_2	S_3	X_i
0	0	\bar{A}_i	0	0	1
0	1	$\bar{A}_i B_i$	0	1	$\bar{A}_i + B_i$
1	0	$\bar{A}_i \bar{B}_i$	1	0	$\bar{A}_i + B_i$
1	1	0	1	1	\bar{A}_i

函数表达式

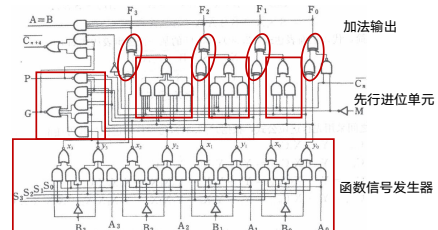
$$X_i = \bar{S}_3 A_i B_i + S_2 A_i \bar{B}_i$$

$$Y_i = A_i + S_0 B_i + S_1 \bar{B}_i$$

47

多功能算数逻辑单元 (ALU)

- 4位ALU芯片74181逻辑电路图（正逻辑）



- M=L/0时，进位信号无影响，算数运算
- M=H/1时，无进位，按位独立，逻辑运算

48

多功能算数逻辑单元 (ALU)

- 算术逻辑运算的实现 (74181)
 - $M=L$ 时, 对进位信号没有影响, 做算术运算
 - $M=H$ 时, 进位门被封锁, 做逻辑运算
- 说明:
 - 74181执行正逻辑输入/输出方式的一组算术运算和逻辑运算和负逻辑输入/输出方式的一组算术运算和逻辑运算是等效的。
 - $A=B$ 端可以判断两个数是否相等。

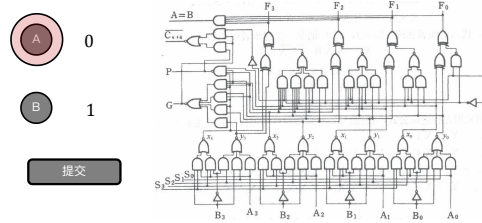
49

单选题 1分

设置

此题未设置答案, 请点击右侧设置按钮

4位74181逻辑电路如下图, 若要实现算术运算, 控制信号M应为

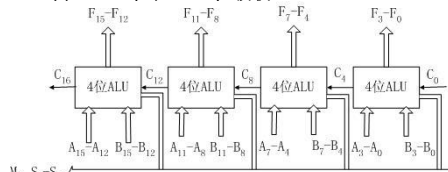


2020-6-23

50

多功能算数逻辑单元的级联

- 设计16位ALU
 - 4片74181 (4位ALU) 级联

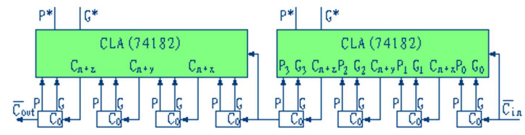


- 片内超前 (先行) 进位, 片间行波进位

51

多功能算数逻辑单元的级联

- 设计32位ALU
 - 8片74181 (4位ALU)
 - 2片超前进位信号发生器 74182

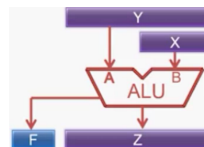


- 16位内超前进位, 16位间行波进位

52

总结

- 逻辑运算器
 - 非、与、或、异或
 - 逻辑运算器实现
- 算数运算
 - 全加器
 - 行波进位加法器、超前进位加法器
 - 加/减运算
- 算数逻辑单元
 - 函数发生器、74181算数逻辑单元



2020-6-23

53