

# LSP 精选例题集 (Selected Examples V2.1)

## 版本信息

- 版本: 2.1
  - 日期: 2024 年 1 月 7 日
  - 内容: 往年考试精选例题, 附详细解答
- 

## 一、有符号/无符号运算

### 例 1: 10 位 signed 加法 (400+500)

题目: signal X, Y, S : signed(9 downto 0); X=500, Y=400, 求 S=X+Y [English] signal X, Y, S : signed(9 downto 0); X=500, Y=400, find S=X+Y

三种解法:

#### A. 二进制转换法

400 = 01 1001 0000  
500 = 01 1111 0100  
400+500 = 11 1000 0100 (=900)

最高位为 1 → 负数, 求补码:

取反: 00 0111 1011  
加1: 00 0111 1100 = 124

答案: -124

#### B. 除法转二进制法 900 除以 2, 记录余数得到二进制, 再转补码

#### C. 会计师法则 (最快)

- 10 位 signed 范围: -512 到 511
- 无符号 1023 = 有符号-1
- 无符号 512 = 有符号-512
- 公式: **signed = unsigned - 1024** (当 unsigned > 511 时)
- $900 \rightarrow 900-1024 = -124$

## 练习题

1. 16 位 signed:  $X+Y=0$ ,  $Y=0x70CA \rightarrow X=0x8F36$
  2. 16 位 signed:  $X+Y=10$ ,  $Y=100 \rightarrow X=0xFFA6$
  3. 8 位 signed:  $X+Y=0$ ,  $Y=0x9A \rightarrow X=0x66$
  4. 8 位 signed:  $X+Y=-1$ ,  $Y=50 \rightarrow X=0xCD$
  5. 14 位 signed:  $X+Y=0$ ,  $Y=1234 \rightarrow X=0x3B2E$
- 

## 二、等价逻辑函数 (用卡诺图判断)

方法: 构造卡诺图比较

例 C4: 检查哪些函数等价

```

f1<=(((not B and not A) or (A and D)) and C) or (B and A);
f2<=(((not A or D) and C) or A) and B;
f3<=((C and (not A or D)) or B) and A and not B;
f4<=((not A or D) and C and not B) or (B and A);
f5<=(A and not B) or (A and D) or (B and A);
f6<=(B or C) and (not A or B or D) and (A or not B);

```

答案: f1 f4

例 D4:

```

f1<=(A xor C) or (A and not C);
f4<=(A or C) and (not A or not C);
f6<=(A and not C) or (C and not A);

```

答案: f4 f6 (都是 XOR)

例 E4:

```

f4<=(not A or not C) and (C or A);
f5<=(not C and A) or (not A and C);

```

答案: f4 f5 (都是 XOR)

#### 快速识别 XOR

- SoP 形式:  $(A \text{ and } \neg B) \text{ or } (\neg A \text{ and } B) = A \text{ XOR } B$
  - PoS 形式:  $(A \text{ or } B) \text{ and } (\neg A \text{ or } \neg B) = A \text{ XOR } B$
- 

### 三、De Morgan 定律应用

例 B1: 将 XOR/XNOR 转换为 AND/OR/NOT

$F1 = A \text{ xor } (B \text{ or } C)$

$F1 = (A \text{ and } \neg B \text{ and } \neg C) \text{ or } (\neg A \text{ and } (B \text{ or } C))$

$F2 = \neg(A \text{ and } (B \text{ xor } C))$

$F2 = \neg A \text{ or } ((\neg B \text{ or } C) \text{ and } (B \text{ or } \neg C))$

---

### 四、Shannon 展开

例: 分解函数  $Q = f(A,B,C,D)$

形式:  $Q = CD \cdot f(A,B) + CD \cdot f(A,B) + CD \cdot f(A,B) + CD \cdot f(A,B)$

方法: 1. 将 C,D 替换为 0,0 → f 2. 将 C,D 替换为 0,1 → f 3. 将 C,D 替换为 1,0 → f 4. 将 C,D 替换为 1,1 → f

从锁存器电路提取 Shannon 展开

例: RS 锁存器电路

```

Q := (A   B) . (Q + (B   C))
f  = (A   B) . (B   C) -- Q='0' 时
f  = (A   B)           -- Q='1' 时

```

---

## 五、RS 锁存器仿真

NOR 门 RS 锁存器 vs NAND 门 RS 锁存器

类型	Set	Reset	保持	禁止
NOR	S=1,R=0	S=0,R=1	S=0,R=0	S=1,R=1
NAND	S=0,R=1	S=1,R=0	S=1,R=1	S=0,R=0

注意：NAND 门 RS 锁存器是低电平有效！

例题

A = 1|1|1|0  
B = 0|1|1|1  
C = 0|0|1|1

答案（取决于电路结构）：- 分析每个时间点的输入 - 确定 Set/Reset/保持状态 - 追踪 Q 和 Q̄ 的变化

---

## 六、组合电路分析

例 E1: 6 输入比较器电路

功能分析：- Z = (FDB = ECA) - 相等比较器 - Y = (FDB > ECA) - 大于比较器

卡诺图填写规则：- 只有高位相等时，低位才影响结果 - 按位权重优先级处理

---

## 七、VHDL 代码编写

D 触发器（带异步清零和同步清零）

```
process(clock, aclrn)
begin
  if aclrn='0' then
    q <= '0';
  elsif rising_edge(clock) then
    if sclrn='0' then
      q <= '0';
    else
      q <= d;
    end if;
  end if;
end process;
```

100 位移位寄存器（最简代码）

```
signal reg: std_logic_vector(99 downto 0);
process(clock)
begin
  if rising_edge(clock) then
    if sclrn='0' then
      reg <= (others=>'0');
    else
      reg <= d & reg(99 downto 1);
```

```

    end if;
end if;
end process;
q <= reg(0);

```

### VHDL 代码解读例题

```

-- 3 位移位寄存器 (异步清零)
signal z: std_logic_vector(0 to 2);
process(a, b)
begin
    if a='0' then
        z <= (others=>'0');
    elsif rising_edge(b) then
        z <= c & z(0 to 1);
    end if;
    d <= z(2);
end process;

-- 2 位寄存器 (异步预置)
signal z: std_logic_vector(1 downto 0);
process(d, c)
begin
    if d='0' then
        z <= (others=>'1'); -- 预置为全 1
    elsif rising_edge(c) then
        z <= b & a;
    end if;
    e <= z;
end process;

```

---

## 八、分支预测器

非考点提示：根据 2026 年 1 月考试说明，分支预测器本次不考，可跳过此节。

例：搜索最大值程序

```

int data[] = {0, 1, -2, 3, 4, -5, -6, -7, 8, 9};
int max = INT_MIN;
for (int i = 0; i < 10; i++) {
    if (data[i] > max) max = data[i];
}

```

编译为 do-while 形式：

```

i = 0;
Loop: if (data[i] <= max) goto Next;
      max = data[i];
Next: i++; if(i < 10) goto Loop;

```

分析过程

循环分支 ( $i < 10$ ): - 1 位预测器 (NT): 开始 miss + 结束 miss = 2 次 - 2 位预测器 (WT): 只在结束 miss = 1 次

条件分支 ( $\text{data}[i] > \text{max}$ ):

数据	0	1	-2	3	4	-5	-6	-7	8	9
跳转	N	N	T	N	N	T	T	T	N	N
1 位	hit	hit	miss	miss	hit	miss	hit	hit	miss	hit
2 位	miss	hit	miss	hit	hit	miss	miss	hit	miss	miss

总结: - 1 位预测器: 2(循环) + 4(if) = 6 次 miss - 2 位预测器: 1(循环) + 6(if) = 7 次 miss

---

## 九、Cache 计算

非考点提示: 根据 2026 年 1 月考试说明, Cache 本次不考, 可跳过此节。

### 地址分解

对于 32 位处理器, 地址分解为:

[Tag | Set Index | Block Offset | Byte Offset]  
 $(m-1) \dots (\log_2(\text{sets}) + \log_2(\text{block}) + 2) \dots (\log_2(\text{block}) + 2) \dots (2) \dots (0)$

例: 64 字节 Cache, 地址 0x1234

直接映射, 块长 1 字

- 16 组, 每组 1 行
- 地址: 0001 0010 0011 0100
- Set = 1101 = 13
- Tag = 72

直接映射, 块长 2 字

- 8 组, 每组 1 行
- Set = 110 = 6
- Tag = 72
- Block offset = 1

2 路组相联, 块长 1 字

- 8 组, 每组 2 行
- Set = 101 = 5
- Tag = 145

2 路组相联, 块长 2 字

- 4 组, 每组 2 行
- Set = 10 = 2
- Tag = 145
- Block offset = 1

例: Cache 命中分析

地址序列: 0x4, 0x8, 0xC, 0x48, 0x4, 0x8, 0xC, 0x88, 0x8, 0xC

直接映射结果

地址	结果	说明
0x4	miss	加载 set 0
0x8	miss	加载 set 1(含 0xC)
0xC	<b>hit</b>	与 0x8 同块
0x48	miss	替换 set 1
0x4	<b>hit</b>	set 0 仍有效
0x8	miss	替换 set 1
0xC	<b>hit</b>	与 0x8 同块
0x88	miss	替换 set 1
0x8	miss	替换 set 1
0xC	<b>hit</b>	与 0x8 同块

结果: 6 miss, 4 hit

## 2 路组相联 (LRU) 结果

地址	结果	说明
0x4	miss	set 0, line 0
0x8	miss	set 1, line 0
0xC	<b>hit</b>	与 0x8 同块
0x48	miss	set 1, line 1
0x4	<b>hit</b>	set 0 仍有效
0x8	<b>hit</b>	set 1, line 0
0xC	<b>hit</b>	与 0x8 同块
0x88	miss	替换 LRU line 1
0x8	<b>hit</b>	set 1, line 0
0xC	<b>hit</b>	与 0x8 同块

结果: 4 miss, 6 hit

## 考试必背要点

### 1. 有符号数转换

- 10 位: 范围-512~511,  $1024=0$
- n 位 signed 转换:  $\text{val} > 2^{n-1}-1$  时,  $\text{val} - 2^n$

### 2. XOR 识别

- SoP:  $(A \cdot B) + (\bar{A} \cdot \bar{B})$
- PoS:  $(A+B) \cdot (\bar{A}+B)$

### 3. RS 锁存器

- NOR: 高电平有效
- NAND: 低电平有效

### 4. Cache 公式 本次不考

- 组数 = Cache 大小 / (块大小 × 相联度)
- 地址位: [Tag | Set | Block | Byte]

5. 分支预测 本次不考

- 1 位: 每次循环 2 次 miss
- 2 位: 每次循环 1 次 miss