一、L3：二进制数据格式

1. 课程回顾与基础铺垫

（1）回顾内容

- 课程大纲、基本逻辑门、二进制数。
- 布尔表达式示例及化简：
  原始表达式：$\overline{A+B\overline{C}}+D(\overline{E+\overline{F}})$
  化简过程：通过德摩根定理等规则逐步推导为$(A+B\overline{C})(\overline{D}+E+\overline{F})$。

（2）学习目标

- 理解八进制（Base-8）与十六进制（Base-16）数制。
- 掌握信息到二进制数的编码方法。
- 理解串行与并行数据传输原理。

（3）推荐阅读

书名	版本	章节
Digital Design	第二版	第 1 章 1.1、1.2 节
Digital Electronics: A Practical Approach with VHDL	第九版	第 1 章、第 2 章 2.3 节

2. 数制系统

（1）十六进制（Base-16）

- **核心对应关系**：1 位十六进制对应 4 位二进制，0-9 对应 0000-1001，A-F 对应 1010-1111。
- **转换方法**：
  1. 十进制→十六进制：先转二进制，再按 4 位分组转十六进制。
     示例：$(99)_{10}$→二进制`01100011`→分组`0110`（6）、`0011`（3）→$(63)_{16}$。
  2. 二进制→十六进制：按 4 位分组（从右向左），每组对应 1 位十六进制。
     示例：$(11110000)_2$→分组`1111`（F）、`0000`（0）→$(F0)_{16}$。
  3. 十六进制→二进制：1 位十六进制拆分为 4 位二进制。
     示例：$(A11)_{16}$→A（1010）、1（0001）、1（0001）→$(101000010001)_2$。

- **实际应用**：RFID 标签 ID 的紧凑表示。32 位 Tag ID 分为 8 位省份码（7→`00000111`→07）、8 位城市码（160→`10100000`→A0）、16 位动物码（513→`0000001000000001`→0201），最终 Hex 格式为`07A00201`。

（2）八进制（Base-8）

- **核心对应关系**：1 位八进制对应 3 位二进制。
- **转换示例**：$(2037)_{10}$→二进制`011111110101`→按 3 位分组（从右向左）`011`（3）、`111`（7）、`110`（6）、`101`（5）→$(3765)_8$。

（3）多进制转换练习

以$(2037)_{10}$为例：

目标数制	转换过程	结果
二进制	对比 2 的幂次（2048、1024...1），确定每一位取值	`011111110101`
十六进制	4 位分组：`0111`（7）、`1111`（F）、`0101`（5）	`0x7F5`
八进制	3 位分组：`011`（3）、`111`（7）、`110`（6）、`101`（5）	`3765`

3. 数据编码为二进制

（1）数字现象编码

- 原理：将非二进制数据（如颜色、符号）映射为二进制串。
- 示例：3 位二进制可表示 8 种状态（000-111 对应 0-7、红 - 黑等）。

（2）BCD 编码（Binary-Coded Decimal）

- **适用场景**：仅用于数值数据编码。
- **规则**：每位十进制数（0-9）对应 4 位二进制数。
  示例：$(17)_{10}$→拆分为 1（0001）和 7（0111）→$(00010111)_{BCD}$。
- **反向转换**：从右向左按 4 位分组，每组转十进制。

（3）ASCII 编码（American Standard Code for Information Interchange）

- **适用场景**：表示字母、数字、符号等 alphanumeric 数据。
- **特点**：采用 7 位编码，共 128 种状态。

- **关键对应关系**：
  - 空格：`0100000`
  - 数字 0-9： `0110000`- `0111001`
  - 大写字母 A-Z：`1000001`（A）-`1011010`（Z）
  - 小写字母 a-z：`1100001`（a）-`1111010`（z）
- **实际应用**：数字温度计，传感器读取温度后，系统生成对应 ASCII 码（0-32→"F"、≥212→"B"、其他→"N"）并输出到 LCD 显示。

（4）模拟信号到二进制编码

- **核心器件**：ADC（模数转换器）、DAC（数模转换器）。
- **ADC 转换三步骤**：
  1. 采样：在固定时间点采集信号（如 1ms、2ms 等）。
  2. 量化：将信号幅值划分等级，将采样值归整到最近等级。
  3. 编码：将量化结果映射为二进制。
     示例：0V→"00"、1V→"01"、2V→"10"、3V→"11"，采样值 1V→"01"、3V→"11"。
- **实际应用**：CD 存储，模拟音频经 ADC 转换为二进制，以反射点（Land）和凹陷（Pit）的变化存储，激光检测反射强度变化生成 0/1 信号。

4. 二进制数据传输

（1）串行传输

- **原理**：单根导线传输，每个时钟周期发送 1 位，从最低有效位（LSB）开始，位变化发生在时钟下降沿。
- **特点**：成本低、速度慢。
- **典型接口及速率**：
  - COM 口：160 kbps
  - USB（1.0/2.0/3.0）：数百 Mbps
  - 以太网口：100 Mbps
- **时序图**：以时钟信号（周期性高低振荡）为基准，标注信号随时间的二进制变化（如`01101100`的串行传输时序）。

（2）并行传输

- **原理**：每位二进制对应一根导线，同时传输所有位。
- **特点**：速度快、成本高。
- **典型接口及速率**：
  - 打印机端口（LPT1）：8 位并行，115 kBps
  - SCSI：16 位并行，320 MBps
  - PCI 总线：32 位并行，133 MBps

（3）传输时间计算

- 公式：时钟周期$T_{clock} = 1/f$（f 为时钟频率）；

  串行传输时间$t_{serial} = n \times T_{clock}$（n 为位数）；

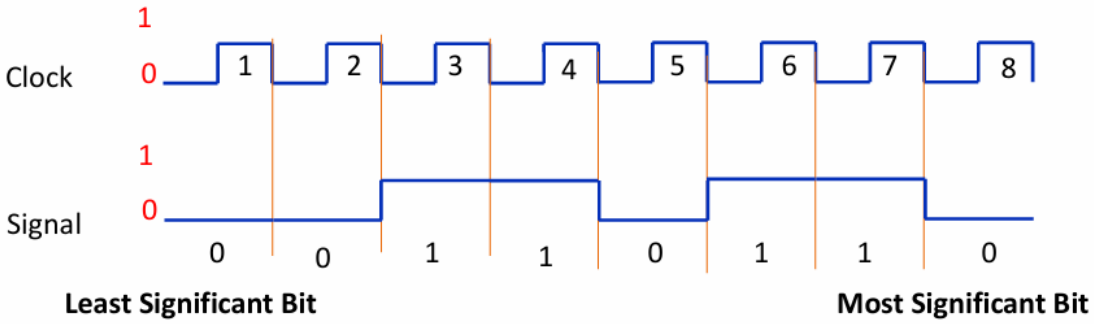
  并行传输时间$t_{parallel} = T_{clock}$。

- 示例：时钟频率 5 MHz（$T_{clock} = 0.2\mu s$），4 位数据（0111）：

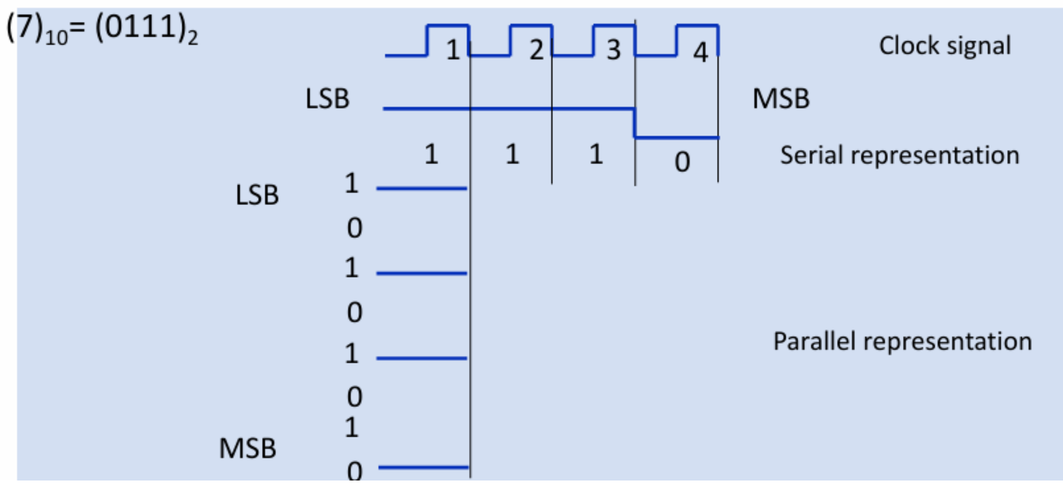
  串行时间 = 4×0.2μs=0.8μs；并行时间 = 0.2μs。

■ How a binary string 01101100 can be transmitted in a serial format?

- One bit per clock period
- Transmission starts from LSB
- Bit-to-bit change occurs at the falling edge



A) Sketch the serial and parallel representation of the decimal number 7 as a 4-bit number.



二、L4：真值表（CNENGR116_L4_Truth Tables.pdf）

1. 课程概述与目标

（1）概述

涵盖布尔表达式与逻辑电路、表达式化简、SOP/POS 形式、SOP/POS 与真值表的转换。

（2）学习目标

- 理解布尔表达式与逻辑电路的对应关系。
- 掌握多变量真值表构建方法。
- 学会布尔表达式化简及 SOP/POS 相关转换。

2. 布尔代数基础

（1）基本规则

1. $A+0=A$
2. $A+1=1$
3. $A\cdot 0=0$
4. $A\cdot 1=A$
5. $A+A=A$
6. $A+\overline{A}=1$
7. $A\cdot A=A$
8. $A\cdot\overline{A}=0$
9. $\overline{\overline{A}}=A$
10. $A+AB=A$
11. $A+\overline{A}B=A+B$
12. $(A+B)(A+C)=A+BC$

（2）德摩根定理

- $\overline{AB}=\overline{A}+\overline{B}$
- $\overline{A+B}=\overline{A}\cdot\overline{B}$

3. 布尔表达式与逻辑电路

- **对应关系**：运算顺序决定电路结构。
  示例：$A(B+CD)$→C 与 D→CD 或 B→结果与 A。

4. 真值表构建

（1）直接法

- 规则：n 个变量→2^n行，逐行计算中间项及结果。

  示例：$A(B+CD)$（4 变量 16 行），依次算 CD、B+CD、A (B+CD)。

（2）逻辑分析法

- 规则：找输出为 1 的条件，直接填充。
  示例：$A(B+CD)=1$需$A=1$且（$B=1$或$C=1$、$D=1$），对应行填 1。

5. 布尔表达式化简

- **示例** 1：
  原始：$AB+A(B+C)+B(B+C)$

  化简：$AB+AC+B+BC=B+AC$。

- **示例** 2：
  原始：$(A\overline{B}(C+BD)+\overline{AB})C$

  化简：$A\overline{B}C+\overline{AB}C=\overline{B}C$。

6. SOP 与 POS 形式

（1）和之积（SOP）

- **定义**：乘积项相加，如$AB+\overline{A}C$。

- **标准形式**：每项含所有变量（原 / 反），如$A\overline{B}C+\overline{A}BC$。

- **二进制意义**：标准项仅对应变量组合为 1，SOP 为 1 当任一乘积项为 1。

（2）积之和（POS）

- **定义**：和项相乘，如$(A+B)(\overline{A}+C)$。
- **标准形式**：每项含所有变量（原 / 反），如$(A+\overline{B}+C)(\overline{A}+B+C)$。
- **二进制意义**：标准项仅对应变量组合为 0，POS 为 0 当任一和项为 0。

（3）SOP 与 POS 转换

- **原理**：有效二进制值互补。
- **步骤**：列 SOP 有效值→找补集→写标准和项→相乘得 POS。

7. 真值表与 SOP/POS 的转换

（1）SOP→真值表

1. 转标准 SOP；2. 找乘积项对应二进制；3. 对应行填 1，其余 0。

（2）POS→真值表

1. 转标准 POS；2. 找和项对应二进制；3. 对应行填 0，其余 1。

（3）真值表→SOP

1. 收集输出为 1 的行；2. 每行写标准乘积项；3. 相加得 SOP。

（4）真值表→POS

1. 收集输出为 0 的行；2. 每行写标准和项；3. 相乘得 POS。

8. 卡诺图（预告）

- **作用**：化简布尔表达式。
- **结构**：3 变量（AB 为行，C 为列）、4 变量（AB 为行，CD 为列）。
- **规则**：相邻 1/0 分组（2^n个单元格），消除互补变量。

要将十进制数 2037_{10} 转换为二进制（base 2）、八进制（base 8）和十六进制（base 16），我们可以分别使用**除基取余法**（转换为二进制、十六进制）和**二进制分组法**（从二进制转八进制）。

一、转换为二进制（Base 2）

使用**除 2 取余法**：将十进制数不断除以 2，记录余数，直到商为 0，最后将余数倒序排列。

1. $2037 \div 2 = 1018$ 余 1
2. $1018 \div 2 = 509$ 余 0
3. $509 \div 2 = 254$ 余 1
4. $254 \div 2 = 127$ 余 0
5. $127 \div 2 = 63$ 余 1
6. $63 \div 2 = 31$ 余 1
7. $31 \div 2 = 15$ 余 1
8. $15 \div 2 = 7$ 余 1
9. $7 \div 2 = 3$ 余 1
10. $3 \div 2 = 1$ 余 1
11. $1 \div 2 = 0$ 余 1

将余数**从下到上**排列，得到：

$2037_{10} = \boxed{11111100101_2}$

二、转换为八进制（Base 8）

先将二进制数**从右往左每 3 位分组**（不足 3 位时左侧补 0），再将每组二进制数转换为对应的八进制数。

1. 二进制 11111100101 分组：
   - 补 0 后：$011\ 111\ 100\ 101$
2. 每组转换为八进制：
   - $011_2 = 3_8$
   - $111_2 = 7_8$
   - $100_2 = 4_8$
   - $101_2 = 5_8$

组合后得到：

$$2037_{10} = \boxed{3745_8}$$

三、转换为十六进制（Base 16）

使用**除 16 取余法**：将十进制数不断除以 16，记录余数（余数对应十六进制符号：0 - 9, A - F），直到商为 0，最后将余数倒序排列。

1. $2037 \div 16 = 127$ 余 5（对应十六进制 5）
2. $127 \div 16 = 7$ 余 15（对应十六进制 F）
3. $7 \div 16 = 0$ 余 7（对应十六进制 7）

将余数**从下到上**排列，得到：
$2037_{10} = \boxed{7F5_{16}}$

最终结果

- 二进制：$\boxed{11111100101_2}$
- 八进制：$\boxed{3745_8}$
- 十六进制：$\boxed{7F5_{16}}$

要化简逻辑表达式 $(A\overline{B}(C+BD)+\overline{AB})C$，我们可以依据**逻辑代数的分配律**、**结合律**和**吸收律**逐步操作。

步骤 1：展开括号（分配律）

根据分配律 $X(Y+Z)=XY+XZ$，对 $A\overline{B}(C+BD)$ 展开：

$A\overline{B}(C+BD)=A\overline{B}C+A\overline{B}BD$

此时原表达式变为：
$(A\overline{B}C+A\overline{B}BD+\overline{AB})C$

步骤 2：化简 $A\overline{B}BD$（同一律、互补律）

根据**同一律** $X\cdot X=X$ 和**互补律** $B\cdot\overline{B}=0$：

- $\overline{B}B=0$，因此 $A\overline{B}BD=A\cdot 0\cdot D=0$

此时表达式简化为：
$(A\overline{B}C+0+\overline{AB})C=(A\overline{B}C+\overline{AB})C$

步骤 3：再次展开括号（分配律）

根据分配律 $(X+Y)Z=XZ+YZ$，展开 $(A\overline{B}C+\overline{AB})C$：

$A\overline{B}C\cdot C+\overline{AB}\cdot C$

步骤 4：化简 $A\overline{B}C\cdot C$（同一律）

根据**同一律** $X\cdot X=X$，$C\cdot C=C$，因此：

$A\overline{B}C\cdot C=A\overline{B}C$

此时表达式简化为：
$A\overline{B}C+\overline{AB}C$

步骤 5：提取公因子（分配律逆用）

提取公因子 $\overline{B}C$：

$\overline{B}C(A+\overline{A})$

步骤 6：化简 $A+\overline{A}$（互补律）

根据**互补律** $X+\overline{X}=1$，$A+\overline{A}=1$，因此：

$\overline{B}C\cdot 1=\overline{B}C$

最终化简结果

$\boxed{\overline{B}C}$

要分析或化简逻辑表达式 $\boxed{\overline{\overline{A+B\overline{C}}+D\overline{(E+\overline{F})}}}$，可通过**德摩根定律**和 ** 双重非（$\overline{\overline{X}}=X$）** 逐步拆解：

步骤 1：应用德摩根定律（对最外层的 "非" 拆分）

德摩根定律：$\overline{X+Y}=\overline{X}\cdot\overline{Y}$（"或非" 等价于 "非与"）。

对整体 $\overline{\overline{A+B\overline{C}}+D\overline{(E+\overline{F})}}$ 应用定律，得：

$\overline{\overline{A+B\overline{C}}}\cdot\overline{D\overline{(E+\overline{F})}}$

步骤 2：处理 "双重非"

根据 "双重非抵消"：$\overline{\overline{X}}=X$，因此 $\overline{\overline{A+B\overline{C}}}=A+B\overline{C}$。

步骤 3：再次应用德摩根定律（拆分 $\overline{D\overline{(E+\overline{F})}}$）

对 $\overline{D\cdot\overline{(E+\overline{F})}}$ 应用德摩根定律：$\overline{X\cdot Y}=\overline{X}+\overline{Y}$（"与非" 等价于 "非或"），得：

$\overline{D}+\overline{\overline{(E+\overline{F})}}$

再处理 "双重非"：$\overline{\overline{(E+\overline{F})}}=E+\overline{F}$，因此这部分变为：

$\overline{D}+E+\overline{F}$

步骤 4：合并结果

将步骤 2 和步骤 3 的结果相乘（逻辑 "与"），最终化简为：

$\boxed{(A+B\overline{C})\cdot(\overline{D}+E+\overline{F})}$

若需要**展开式**，可通过 "分配律"（$X\cdot(Y+Z)=X\cdot Y+X\cdot Z$）进一步展开：

$A\overline{D}+AE+A\overline{F}+B\overline{C}\overline{D}+B\overline{C}E+B\overline{C}\overline{F}$

在数字电路（数电）中，SOP 和 POS 是两种**标准的逻辑函数表达式形式**，二者可通过 “摩根定律” 和 “真值表” 实现相互转换，核心是利用逻辑代数的基本规则处理原函数与反函数的关系。

一、先明确核心概念

- **SOP（Sum of Products，积之和）**：由多个 “与项（乘积项）” 进行 “或运算” 组成的表达式。
  例：$F = AB + \overline{A}C + B\overline{C}$（“与项” 指变量或其反变量的乘积，如$AB$；“和” 指或运算）。
- **POS（Product of Sums，和之积）**：由多个 “或项（和项）” 进行 “与运算” 组成的表达式。
  例：$F = (A+B)(\overline{A}+C)(B+\overline{C})$（“或项” 指变量或其反变量的和，如$A+B$；“积” 指与运算）。

二、SOP 与 POS 的转换方法（以 “从 SOP 转 POS” 为例，反向同理）

方法 1：利用 “反函数” 与摩根定律（通用方法）

1. **写出 SOP 形式的原函数F**，先求其反函数$\overline{F}$；
   - 规则：对 SOP 中的 “与” 变 “或”、“或” 变 “与”，每个变量取反（摩根定律）。
     例：若$F = AB + \overline{A}C$（SOP），则$\overline{F} = \overline{AB + \overline{A}C} = \overline{AB} \cdot \overline{\overline{A}C} = (\overline{A}+\overline{B})(A+\overline{C})$（$\overline{F}$为 POS 形式）。
2. **对$\overline{F}$再次取反**，得到原函数的 POS 形式；
   - 规则：对$\overline{F}$（POS）再次应用摩根定律，“与” 变 “或”、“或” 变 “与”，每个变量取反。
     例：$F = \overline{\overline{F}} = \overline{(\overline{A}+\overline{B})(A+\overline{C})} = \overline{\overline{A}+\overline{B}} + \overline{A+\overline{C}} = (A+B)(\overline{A}+C)$（$F$的 POS 形式）。

方法 2：利用真值表（直观易懂，适合变量少的情况）

1. **列出原函数F的真值表**，标记出$F=1$和$F=0$的所有输入组合；
2. **SOP 转 POS**：
   - SOP 由$F=1$的输入组合对应 “与项” 相加得到；
   - POS 由$F=0$的输入组合对应 “或项” 相乘得到（每个$F=0$的组合中，变量为 0 则写原变量，为 1 则写反变量，组成或项）。
     例：$F = AB + \overline{A}C$（SOP，对应$F=1$的组合），其$F=0$的组合为（A=0,B=0,C=0）、（A=0,B=1,C=0）、（A=1,B=0,C=1）、（A=1,B=1,C=0），对应的或项为$(A+B+C)$、$(A+\overline{B}+C)$、$(\overline{A}+B+\overline{C})$、$(\overline{A}+\overline{B}+C)$，故 POS 形式为
     $F = (A+B+C)(A+\overline{B}+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)$。

真值表其实是 “把所有输入情况列全，对应出输出结果” 的表格，核心是 “穷尽所有可能，直观看输入和输出的关系”，用一个简单例子拆解开你就懂了。

先拿 “2 变量逻辑函数 F=A+B（或运算）” 举例，一步一步建真值表

第一步：确定 “输入变量” 和所有可能的组合

逻辑变量只有 0 和 1 两种状态，若有 n 个变量，输入组合共 2^n种（2 个变量就是 2^2=4 种，3 个变量就是 8 种）。
以 2 变量 A、B 为例，所有输入组合按顺序列全：

- 组合 1：A=0，B=0
- 组合 2：A=0，B=1
- 组合 3：A=1，B=0
- 组合 4：A=1，B=1

第二步：根据逻辑函数，算出每种输入对应的 “输出 F”

函数是 F=A+B（或运算：只要 A、B 有一个为 1，F 就为 1；都为 0 时 F 才为 0），所以：

- 组合 1（A=0,B=0）：F=0+0=0
- 组合 2（A=0,B=1）：F=0+1=1
- 组合 3（A=1,B=0）：F=1+0=1
- 组合 4（A=1,B=1）：F=1+1=1

第三步：整理成表格，就是真值表

输入 A	**输入** B	**输出** F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

再回到 “SOP 转 POS” 里的真值表用法（用 3 变量函数 F=AB+¬AC 举例）

第一步：列全 3 变量（A、B、C）的所有输入组合（共 2^3=8 种）

第一步：列全 3 变量（A、B、C）的所有输入组合（共 2^3=8 种）

序号	输入 A	输入 B	输入 C
1	0	0	0
2	0	1	0
3	0	0	1
4	0	1	1
5	1	0	0
6	1	1	0
7	1	0	1
8	1	1	1

第二步：算每种组合的输出 F（F=AB+¬AC，“¬A”是 A 的反，A=0 时 ¬A=1，A=1 时 ¬A=0）

- 序号 1（A=0,B=0,C=0）：AB=0×0=0；¬AC=1×0=0 → F=0+0=0
- 序号 2（A=0,B=1,C=0）：AB=0×1=0；¬AC=1×0=0 → F=0
- 序号 3（A=0,B=0,C=1）：AB=0×0=0；¬AC=1×1=1 → F=1

- 序号 3（A=0,B=0,C=1）：AB=0×0=0；¬AC=1×1=1 → F=1

- 序号 4（A=0,B=1,C=1）：AB=0×1=0；¬AC=1×1=1 → F=1
- 序号 5（A=1,B=0,C=0）：AB=1×0=0；¬AC=0×0=0 → F=0
- 序号 6（A=1,B=1,C=0）：AB=1×1=1；¬AC=0×0=0 → F=1
- 序号 7（A=1,B=0,C=1）：AB=1×0=0；¬AC=0×1=0 → F=0
- 序号 8（A=1,B=1,C=1）：AB=1×1=1；¬AC=0×1=0 → F=1

第三步：标记 F=1 和 F=0 的组合，对应 SOP 和 POS

- **SOP（积之和）**：只看 F=1 的组合，每个组合对应一个“与项”（变量为 1 写原变量，为 0 写反变量）。
  比如 F=1 的是序号 3（A=0,B=0,C=1）→ ¬A¬BC；序号 4→¬ABC；序号 6→ABC；序号 8→ABC，所以 SOP 就是 F=¬A¬BC + ¬ABC + ABC + ABC（可化简为之前的 AB+¬AC）。
- **POS（和之积）**：只看 F=0 的组合，每个组合对应一个“或项”（变量为 0 写原变量，为 1 写反变量）。
  比如 F=0 的是序号 1（A=0,B=0,C=0）→ A+B+C；序号 2→A+¬B+C；序号 5→¬A+B+C；序号 7→¬A+B+¬C，所以 POS 就是 F=(A+B+C)(A+¬B+C)(¬A+B+C)(¬A+B+¬C)。