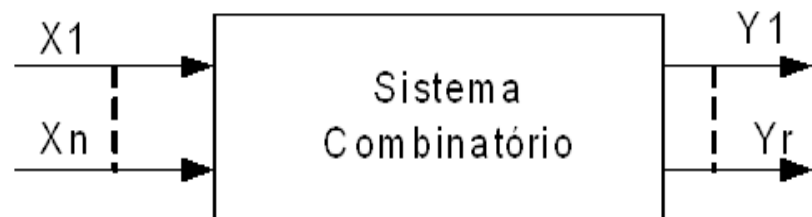


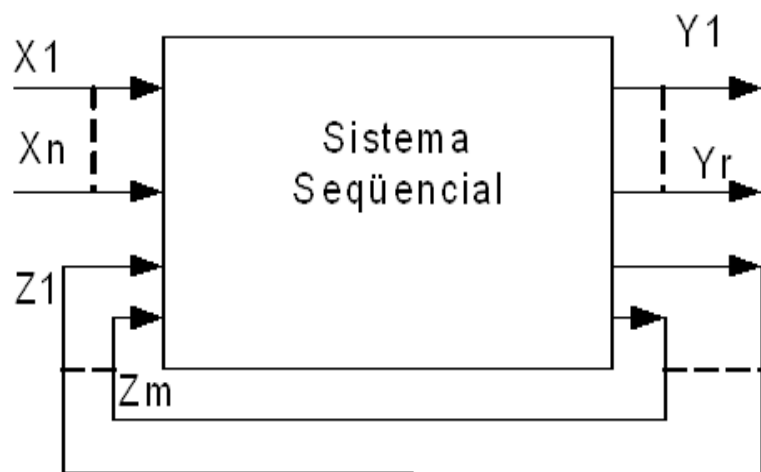


Sistema Digital



Sistema digital → composto por um ou dois tipos:

- a) Combinatório** → Os valores das saídas somente dependem da combinação dos valores das entradas.
- b) Sequencial** → Os valores das saídas dependem da combinação dos valores das entradas primárias e secundárias (**eventos passados**).



Variáveis Booleanas □ X, Y e Z
Sistema Seqüencial □ Entradas primárias X_i
Saídas primárias Y_i
Variáveis secundárias Z_i

Sistema Combinatório



Especificação Formal:

- a) Tabela Verdade
- b) Diagrama de Temporização
- c) Linguagem de Descrição de Hardware
por exemplo: VHDL e Verilog
- d) Funções Booleanas

Problema: *Especificação informal* → Sintetize um circuito combinatório que verifica se um número de 3 bits é ímpar (saída $F=1$) ou par (saída $F=0$).

Spec: Tabela Verdade



Problema: Especifique em uma tabela verdade (TV) um circuito combinatório que verifica se um número de 3 bits é ímpar (saída $F=1$) ou par (saída $F=0$).

A_3	A_2	A_1	F	
0	0	0	0	→ Maxtermo
0	0	1	1	
0	1	0	0	
0	1	1	1	→ Mintermo
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	1	

A TV é de simples saída especificada completamente



Espec: Diagrama de Temporização

Problema: Especifique em um Diagrama de Temporização (DT) um circuito combinatório que verifica se um número de 3 bits é ímpar (saída $F=1$) ou par (saída $F=0$).

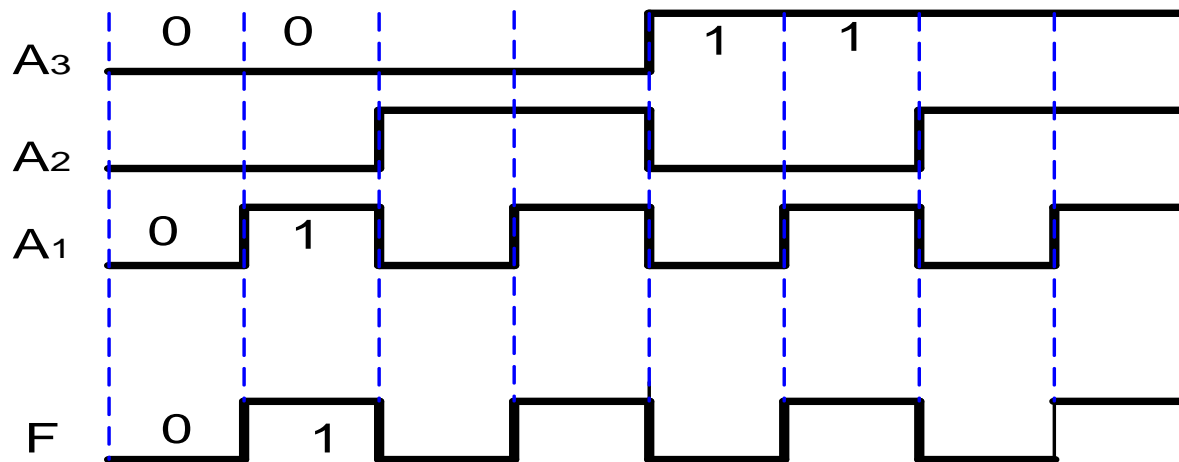


Diagrama de Temporização → Ideal



Espec: Função Booleana

Problema: Especifique uma Função Booleana (**FB**) que verifica se um número de 3

Tabela
Verdade

bits é impar (saída $F=1$) ou par (saída $F=0$).

A_3	A_2	A_1	F
0	0	0	0 → Maxtermo
0	0	1	1
0	1	0	0
0	1	1	1 → Mintermo
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

a) **Mintermo:** Termo produto em que a variável aparece exatamente uma vez $\rightarrow A'_3 A'_2 A_1$ (variável de saída é 1). As entradas com valor zero são complementadas

b) **Maxtermo:** Termo soma em que a variável aparece exatamente uma vez $\rightarrow A_3 + A_2 + A_1$ (variável de saída é 0). As entradas com valor hum são complementadas

Espec: Função Booleana



Problema: Especifique uma **FB** que verifica se um número de 3 bits é ímpar (saída $F=1$) ou par (saída $F=0$).

Uma forma de descrever uma expressão Booleana:

a) **Função Canônica** soma de produto (SOP)

A ₃	A ₂	A ₁	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$F(A_3, A_2, A_1) = A'_3 A'_2 A_1 + A'_3 A_2 A_1 + A_3 A'_2 A_1 + A_3 A_2 A_1$$

b) **Função Canônica** produto da soma (POS)

$$F(A_3, A_2, A_1) = (A_3 + A_2 + A_1) \cdot (A_3 + A'_2 + A_1) \cdot (A'_3 + A_2 + A_1) \cdot (A'_3 + A'_2 + A_1)$$

Função Canônica SOP → contém o máximo número de termos produto e cada termo produto contém o máximo número de **literais** (envolve todas as variáveis da função)

Literal → é uma variável ou o seu complemento



Espec: Função Booleana

Especifique uma FB: Designação decimal

a) **Função Canônica** soma de produtos SOP)

$$F(A_3, A_2, A_1) = \sum (1, 3, 5, 7)$$

b) **Função Canônica** produto das somas (POS)

$$F(A_3, A_2, A_1) = \prod (0, 2, 4, 6)$$

A ₃	A ₂	A ₁	F
0	0	0	0 → Maxtermo
0	0	1	1
0	1	0	0
0	1	1	1 → Mintermo
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Espec: Linguagem de Descrição de Hardware



```
Library ieee;
```

```
use ieee.std_logic_1164 all;
```

```
entity Impar is
```

```
port ( A3, A2, A1 : in std_logic;
```

```
       F : out std_logic
```

```
);
```

```
end Impar;
```

```
ARCHITECTURE Par of Impar is
```

```
Begin
```

```
    If A1='1' then F <= '1' ;
```

```
        else F <='0';
```

```
    end if;
```

```
end Par;
```

Especifique em uma LDH um circuito combinatório que verifica se um número de 3 bits é impar (saída F=1) ou par (saída F=0).

Por exemplo: VHDL