

## F541 Aula 1

### Sinais analógicos e digitais, sistema decimal de numeração, sistemas de bases diferentes.

#### 1-INTRODUÇÃO:

A interação e a compreensão de qualquer processo físico, implica em medições de grandezas físicas que caracterizam este processo. As grandezas físicas medidas constituem uma curva,  $f(t)$ , geralmente contínua, cuja intensidade é um número decimal associado a uma unidade de medida. Exemplos são os sinais fornecidos por voltímetros, termômetros, detectores de intensidade de luz, medidores de acidez (pH-metros), pressão, etc.

A informatização deste processo permite que analisemos o processo em um computador. Para isso temos que transformar as medidas analógicas, em medidas digitais.

Para a digitalização da curva  $f(t)$  a intensidade do sinal deve ser "amostrada" e "segurada" (sample and hold circuit) por um tempo durante o qual é convertida à base dois e armazenada na memória do processador, repetindo-se o processo a cada amostragem do sinal. O sistema binário é usado porque fornece um grande número de vantagens associadas à lógica, armazenamento, processamento de sinais, transmissão, etc.

O sistema binário, octal e hexadecimal.

Conversão de um número decimal em binário.

Um número decimal é convertido em binário dividindo-o por 2 consecutivamente e juntando o quociente final aos restos consecutivos das divisões, ex.  $13/2=6$  sobra 1.  $6/2=3$  sobra 0.  $3/2=1$  sobra 1. O número será 1101; ou seja  $13_{10}=1101_2$ . Para encontrar o valor decimal do binário basta fazer as potências de 2:  $1101_2=1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13$

Existem algumas codificações de binários que devemos conhecer.

1) **Representação hexadecimal do binário.** O número em hexadecimal tem 16 algarismos.

Usa a representação binária de 4 em 4 casas binárias. Associa um dos 16 números: de 0 a 9 depois A, B, C, D, E, F, (equivalente aos números 10, 11, 12, 13, 14, 15 respectivamente).

Ex:  $59_{10}=111011_2=3B_{16}$

2) **Representação octal do binário.**

Usa a representação binária de 3 em 3 casas. Ex:  $59_{10}=111011_2=73_8$

3) **Codificação BCD** ou Binary-coded decimal.

Cada grupo de 4 bits representa um número decimal, de 0 a 9.

Ex:  $01011001_{BCD}=59_{10}$

#### Ilustração de operações com binários:

*Números negativos.*

Para fazer a representação binária de um número negativo, o código mais usado é o complemento de 2 mais um. Exemplos:

$7=0111$  então  $-7=1000+1=1001$ . Observe que  $7+(-7)=0$  basta somar  $0111+1001=0000$ , vai um.

$32=00100000$  portanto  $-32=11011111+1=11100000$

Portanto  $00100000+$

11100000

00000000

Somas e multiplicações são feitas como com decimais

$0001 * 0010 = 0010$ ;

$0011 * 0010 = 0110$ ;

$0011 * 0011 = 0011 + 0110 = 1001$ .

Os números 1 e 0 em cada casa binária são associados a Estados lógicos: High = Verdadeiro=1; Low= Falso=0. Veja o desenho, Fig. 1.

Por exemplo, Um conjunto de variáveis lógicas (digitais), A, B, C, D, pode refletir uma função lógica  $F(A, B, C, D)$ . Os valores

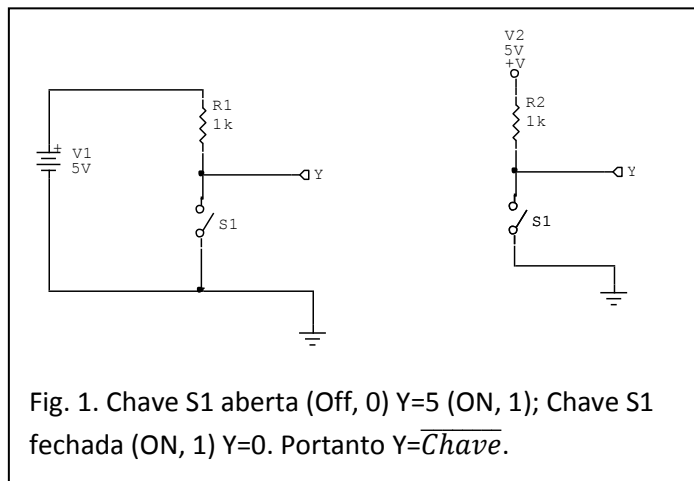


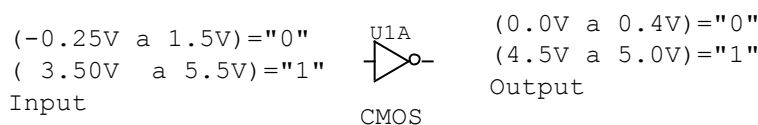
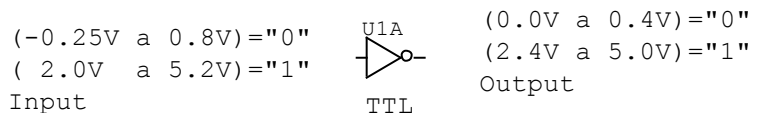
Fig. 1. Chave S1 aberta (Off, 0)  $Y=5$  (ON, 1); Chave S1 fechada (ON, 1)  $Y=0$ . Portanto  $Y=\overline{\text{Chave}}$ .

das variáveis e das funções *só podem ser 1 ou 0*.

## 2- Famílias Lógicas .

Para executar as funções lógicas, as tecnologias mais conhecidas são: TTL (transistor-transistor-logic) e CMOS. (Complementar Metal oxide semiconductor, (complem.Field effect transistors)). A TTL foi uma das primeiras tecnologias desenvolvidas. Os chips TTL são mais rápidos, mas gastam mais energia, relativamente á técnica CMOS. A CMOS é a tecnologia atualmente usada. Os chips CMOS ficaram mais rápidos e econômicos.

Os chips de ambas as tecnologias podem ser alimentados com o mesmo valor de tensão, 5.0V. Porém, apresentam 1s e 0s em diferentes tolerâncias de valores, Veja abaixo:



Note que a imunidade ao ruído de tensão do CMOS é maior do que a do TTL. Porém o ruído em circuitos digitais, também depende de outros fatores como taxa de subida e descida de sinais, estabilidade, etc.

## 3- Portas lógicas (Gates)

A funções mais comuns são: AND, OR, NOT, EXOR, NAND, NOR, conforme a Tabela abaixo.

Lembre que nas operações de OR (ou, soma)  $C = A + B$ , AND (interseção, produto)  $D = A \cdot B$ , NOT (inversão, complemento)  $E = \bar{D}$ . Os elemento A, B, C, D, E só podem valer 1 ou 0.

A operação EXOR é conhecida como meia soma.

A	B	AND	NAND	OR	NOR	EXOR
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

Para a construção das funções lógicas usamos a algebra de Boole e as leis de De Morgan, discutidas em classe.

### Experimental:

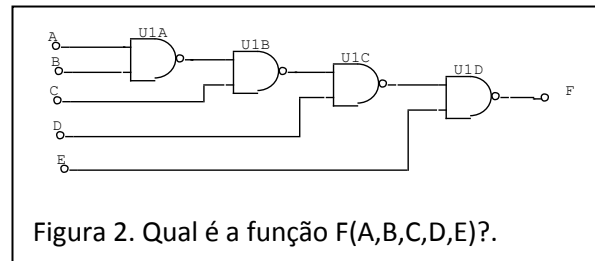
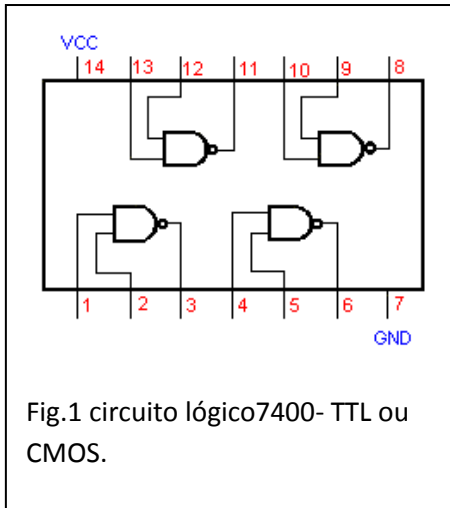
**1- Procure na internet pelo manual do IC 74C00**, um CMOS, que copia o TTL-7400. Coloque num proto-board e alimente-o com 5V.

Utilize o manual para identificar os pinos de alimentação e das portas NAND.

### 4- Monte a função lógica da Figura 2 usando o TTL 7400 da Figura 1.

Use um LED com uma resistência de 150Ω em série, para identificar o estado lógico das portas/funções. Simule as entradas, conectando-as a GND (0) ou a VCC (1), ou seja, 0,0 ou 5,0 Volts.

Descubra a “tabela verdade” da função F(A,B,C,D,E), Fig. 2. Explique o funcionamento. Mostre a função de saída.



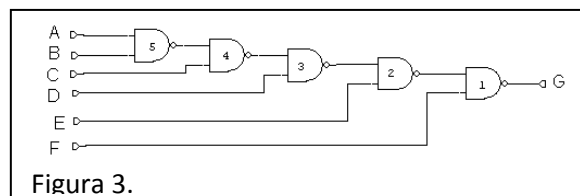
**5- Implemente um AND com 2 NANDs.**

**6- Implemente um EXOR com 4 Nands.**

**7- NANDs de nível par e ímpar.** Na lógica digital combinacional o estado presente da variável de saída é função do estado presente das variáveis de entrada. Conforme a fig. 3

$$\overline{AB} \cdot C \cdot D \cdot E \cdot F = G, \text{ ou:}$$

$((\bar{A} + \bar{B}) \cdot C + \bar{D}) \cdot E + \bar{F} = G$  (use os teoremas de De Morgan.). Observe que os **NANDs de nível par funcionam como ANDs. Que os NANDs de nível ímpar funcionam como OR complementando as entradas.**



**8- Usando NANDs do IC 74C00, implemente o circuito da Figura 4. Mostre a função F(ABCDE)** usando os conceitos de NANDs de nível par e ímpar.

