

Vamos começar com conversões:

$2 \Rightarrow 10$

$$10000010 = 2 + 128 = 130$$

$$1101111 = 2^0 + 2^1 + 2^2 + 2^3 + 2^5 + 2^6 = 47 + 64 = 111$$

$10 \Rightarrow 2$

$$32/2 = 16/2 = 8/2 = 4/2 = 2/2 = 1$$

Portanto, pegando os restos e a última divisão, de trás para frente:

$$32 = 10000$$

$$280/2 = 140/2 = 70/2 = 35/2 = 17/2 = 8/2 = 4/2 = 2/2 = 1$$

Portanto, pegando os restos e a última divisão:

$$280 = 256 + 16 + 8 = 280$$

Podemos também trabalhar com a base 16:

$$AE = 10 + 15 = 25$$

$$A016 = 6 + 11 + 0 + ?$$

$$A00 = 0 \times 16^0 + 0 \times 16^1 + 10 \times 16^2 =$$

Uma porta and ou e é uma porta que exige que todas suas entradas sejam 1. Ou seja, para memorizar, uma porta tem que ser 1 E outra tem que ser 1 etc:

Uma porta OU ou OR exige que apenas 1 entrada seja 1. Ou seja, para memorizar, 1 porta tem que ser 1 ou outra etc:

Uma porta NOT ou Não quando sua entrada for 1, sua saída será 0. E vice versa. É chamada também de porta inversora.

Existem portas derivadas das anteriores:

Estas são a NOR (ou NOT OR) e a NAND (ou NOT AND)

Quando queremos inverter algo colocamos uma barra sobre a expressão.

Circuitos integrados comerciais:

Os circuitos integrados comerciais estão disponíveis como chips CMOS ou TTL (as diferenças entre ambos ficam para outro documento...), mas temos que:

Porta CMOS TTL link

2 and 7408

https://pt.m.wikipedia.org/wiki/Lista_dos_circuitos_integrados_da_série_7400

Como mostra a tabela acima, diferentes CIs dão o mesmo resultado.

A combinação de diferentes CIs podem resultar em uma nova porta lógica. Ex:

Esta é a porta XOR ou exclusive OR. A porta coincidência é a abaixo:

Simplificação de expressões:

As expressões booleanas podem ser simplificadas, como mostram os exemplos abaixo: