

SimulIDE Circuit Simulator

PARTE II - Eletrônica Digital

No segmento Eletrônica Digital, faremos uma análise de instrumentos e componentes que são prioritários no desenvolvimento de projetos e simulações, para este segmento.

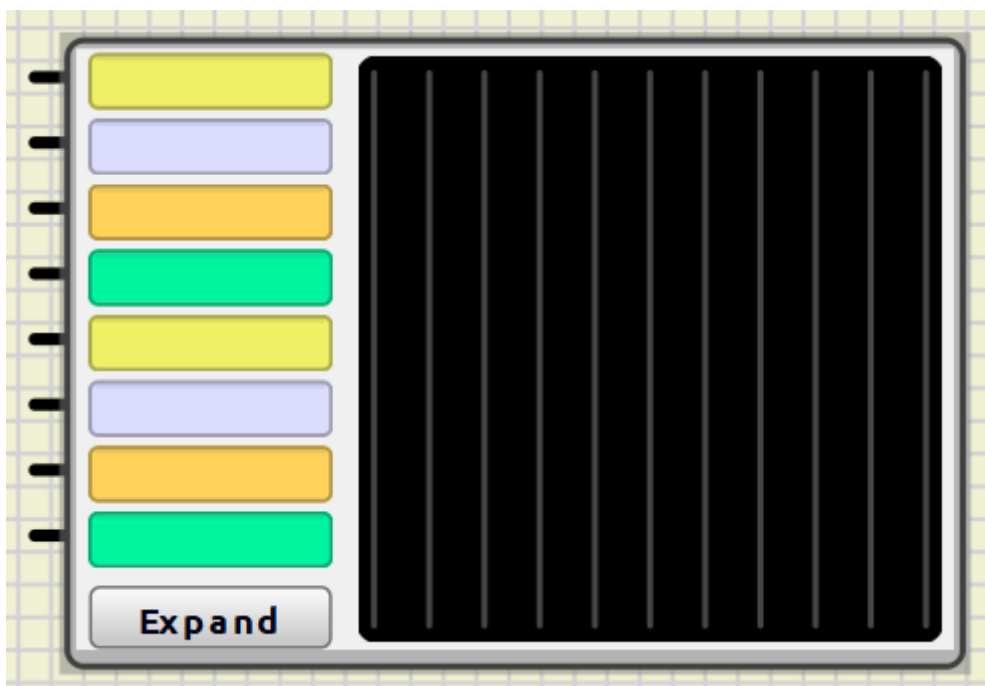
Medidores (Meters)

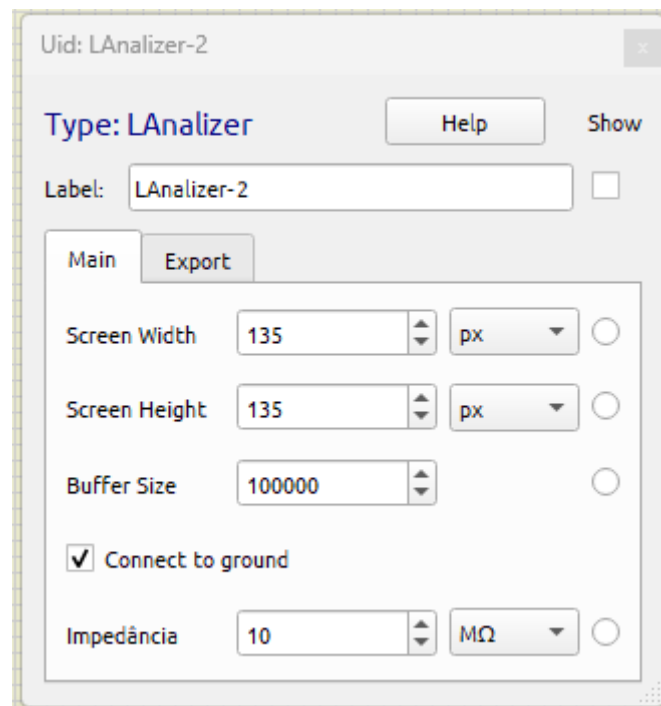
Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou **botão direito do mouse – propriedades.**

Analizador Lógico (Logic Analyzer / LAnalyzer):

A função do analisador lógico (Logic Analyzer) é medir ou capturar e exibir graficamente sinais digitais (níveis lógicos 0 e 1) ao longo do tempo, em múltiplos canais simultâneos.

Ele apenas mede ou monitora os estados do circuito, não processando ou alterando os dados do mesmo, podendo a captura ocorrer através de circuitos osciladores ou de circuitos com chaveamento eletrônico.



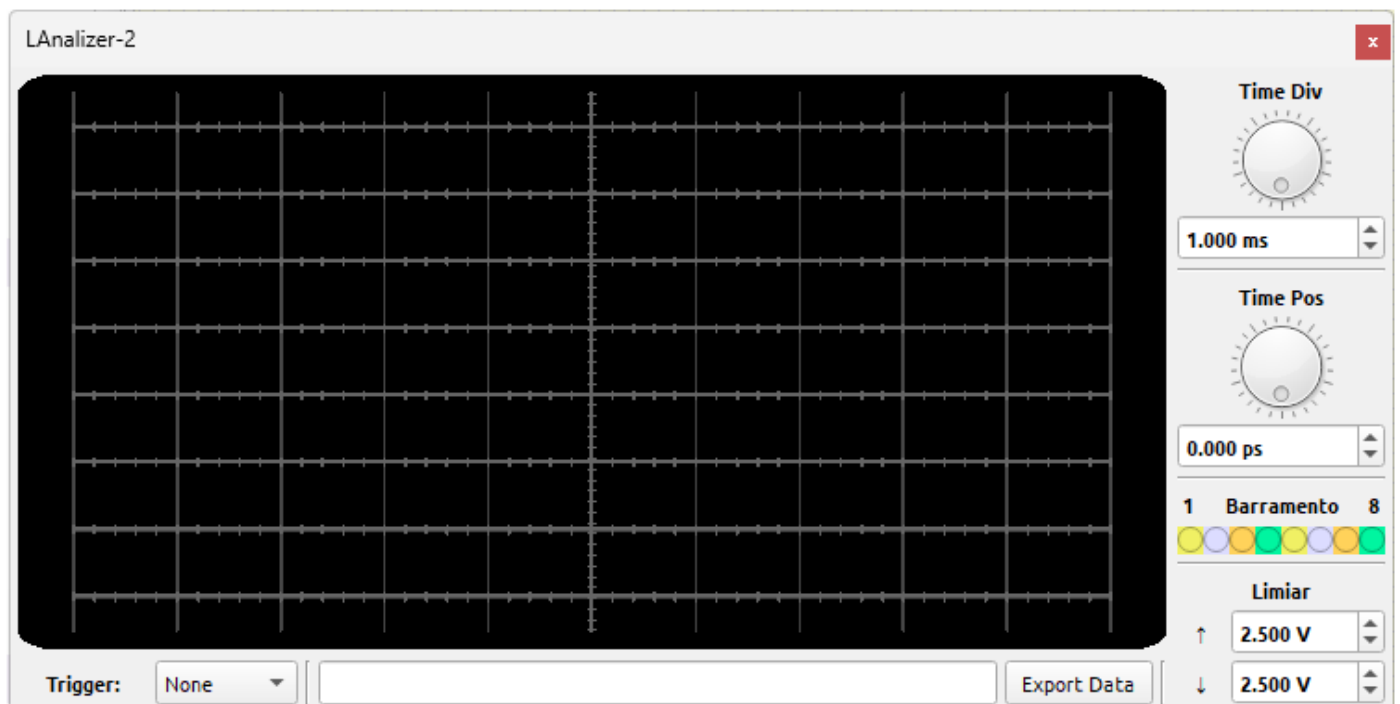


Ao clicar no botão "Expand", é possível ajustar a base de tempo e o posicionamento dos sinais capturados, para uma interpretação adequada da leitura.

A largura e altura de visualização do osciloscópio, podem ser definidos em Screen Width x High (no caso, 135 x 135 px).

Com isso, é possível comparar dois ou mais sinais simultaneamente com muita precisão.

É possível fazer uma comparação simultânea de 8 sinais, visto que o medidor disponibiliza 8 canais.

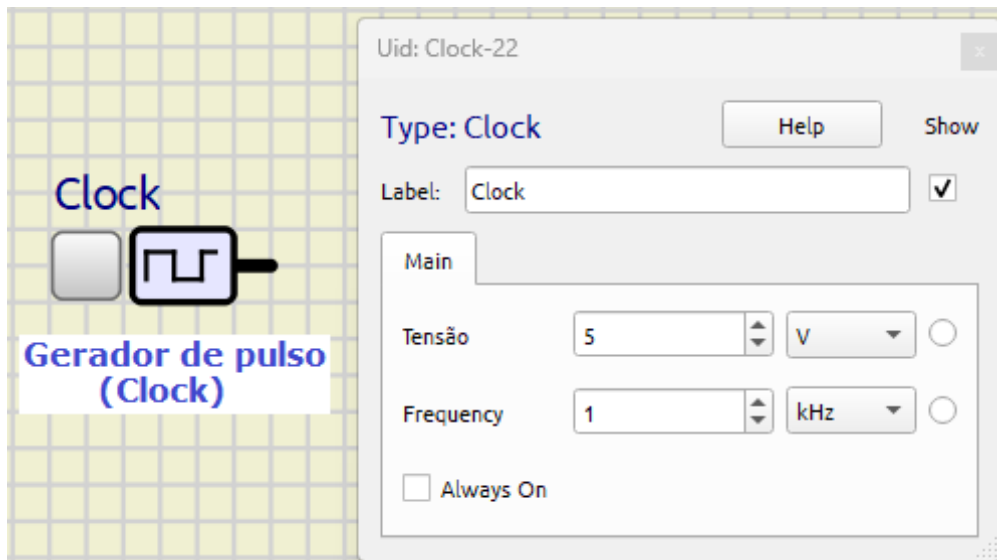


Fontes (Sources)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

Gerador de pulso (Clock):

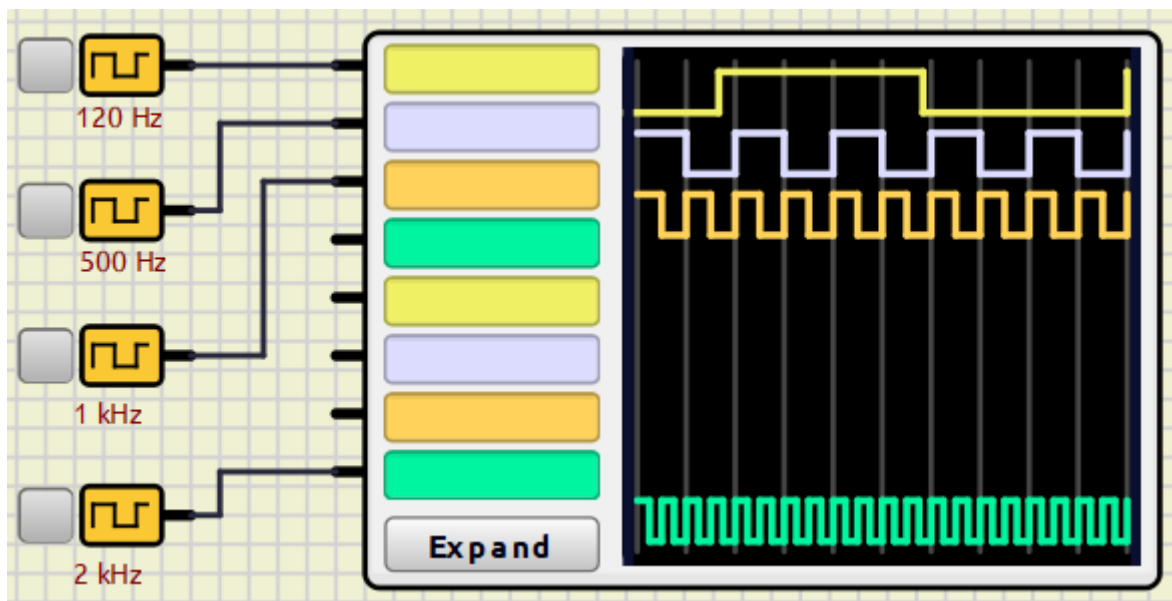
A figura a seguir mostra um gerador de clock, que na realidade trata-se de um oscilador que gera onda quadrada, ou propriamente “pulsos de clock”.



IMPORTANTE: o botão disponível à esquerda do ícone, permite ligar/desligar a função, mesmo com o processo de simulação ativo.

Observa-se que o pulso de clock foi ajustado para 5V de amplitude para uma frequência de 1kHz.

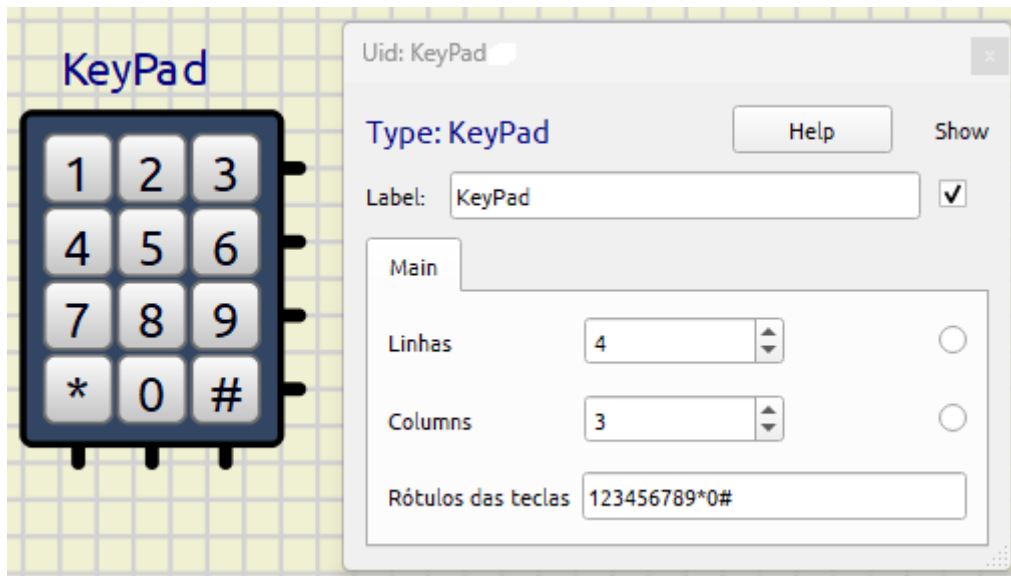
A figura abaixo mostra um Analisador Lógico com diferentes frequências de pulso de clock, nos canais/entradas 1, 2, 3 e 8



Interruptores (Switches)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

Teclado (KeyPad):

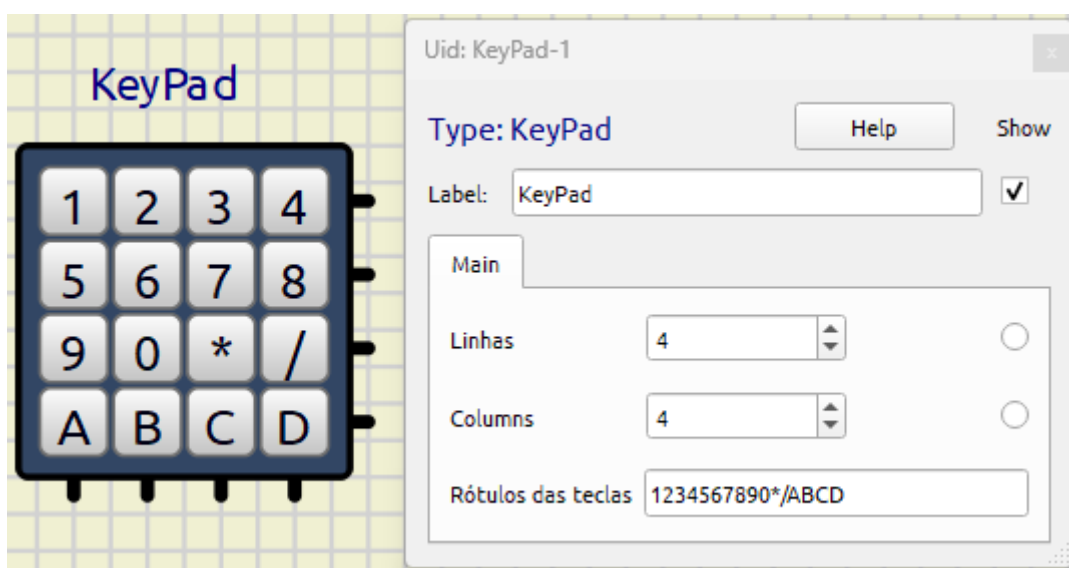


Aplicação mais frequente na simulação de microcontroladores, esse teclado pode ser configurado conforme as necessidades de um projeto.

O teclado mostrado está configurado para 4 linhas e 3 colunas, num total de 12 teclas.

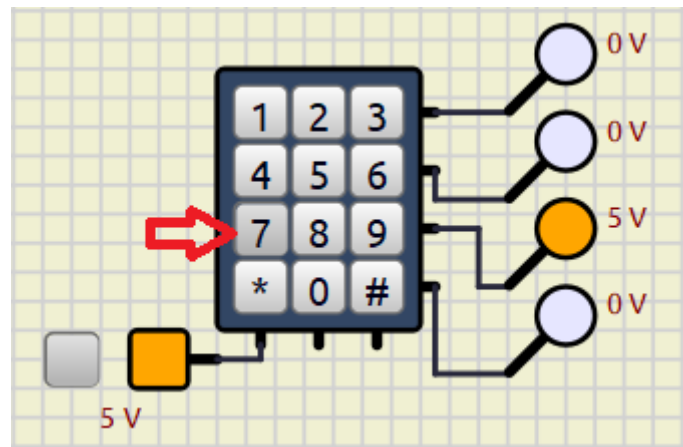
No entanto, de acordo com as necessidades de um projeto existe a possibilidade de alterar o layout do mesmo, em termos de arranjo e colunas, bem como, definir as teclas que serão usadas. A figura a seguir mostra uma configuração matriz tipo 4x4, com 16 teclas e respectiva identificação das mesmas

1234567890*/ABCD



O arranjo mostrado a seguir, mostra a aplicação do teclado em eletrônica analógica. A coluna que corresponde as teclas 1 4 7 * é alimentada por uma tensão de 5 volts.

Ao pressionar qualquer uma dessas teclas (1 ou 4 ou 7 ou *) teremos na saída a tensão correspondente a 5V, que poderá ser utilizada para alimentação de circuitos remotos.



Saídas (outputs)

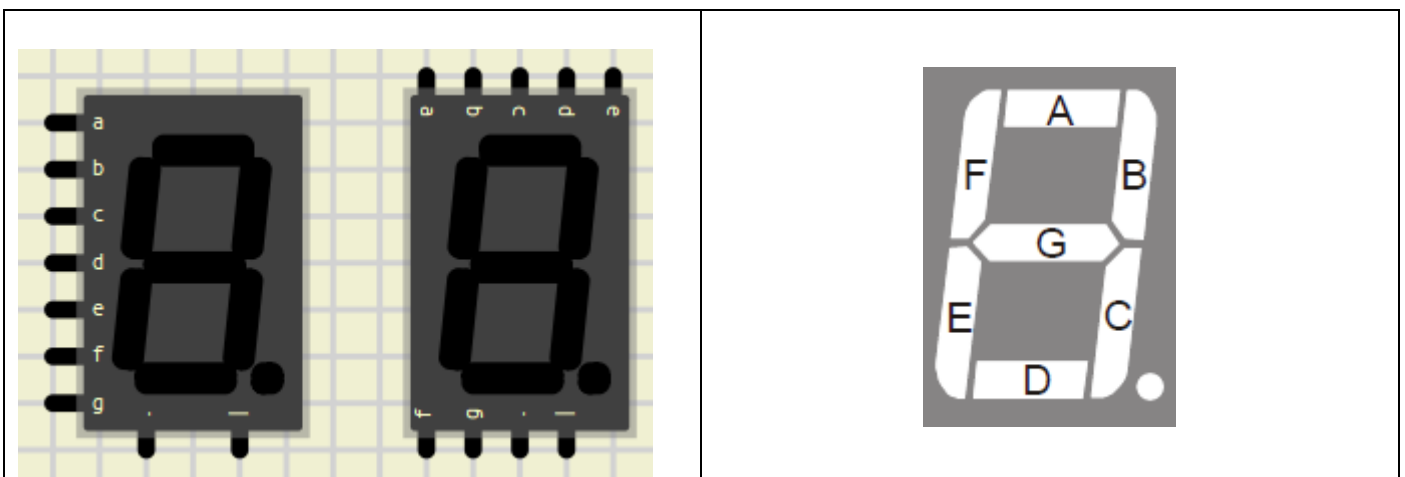
Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

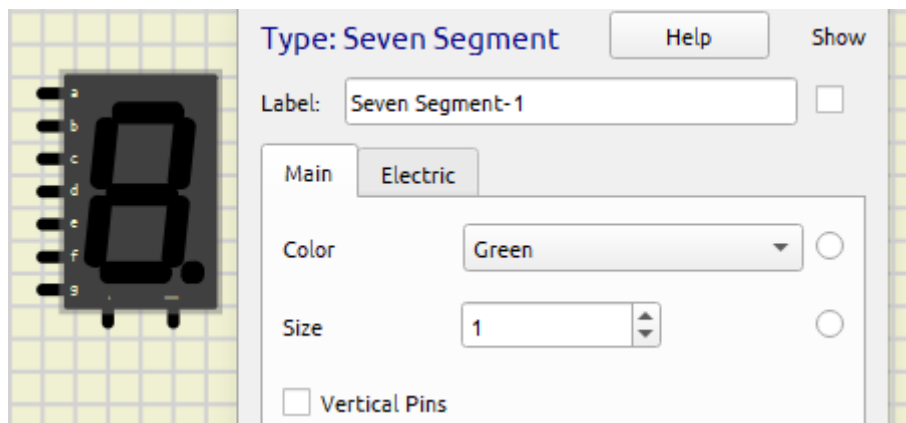
7 segmentos (Seven Segment):

É o display convencional de 7 segmentos, onde cada segmento é representado por um LED.

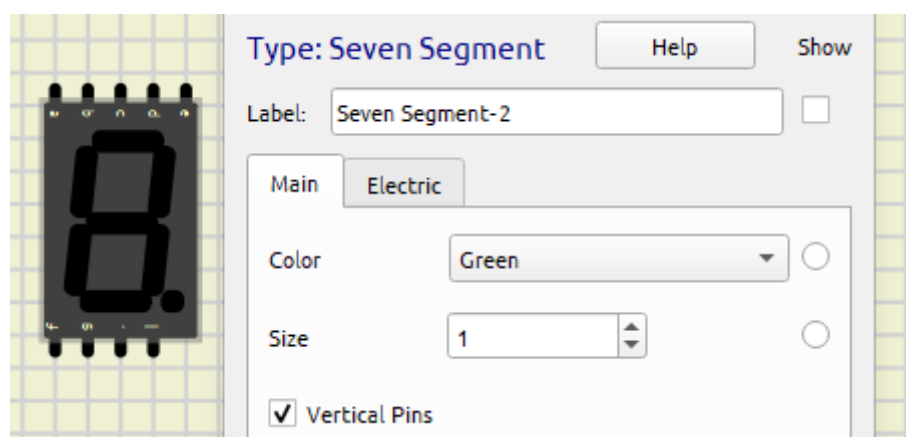
Temos no total 8 leds, sendo 1 para cada segmento (a, b, c, d, e, f, g) e 1 para o segmento "ponto" (DOT).

O simulador permite disponibilizar os pinos tanto na vertical como na horizontal, na opção "Pinos verticais"

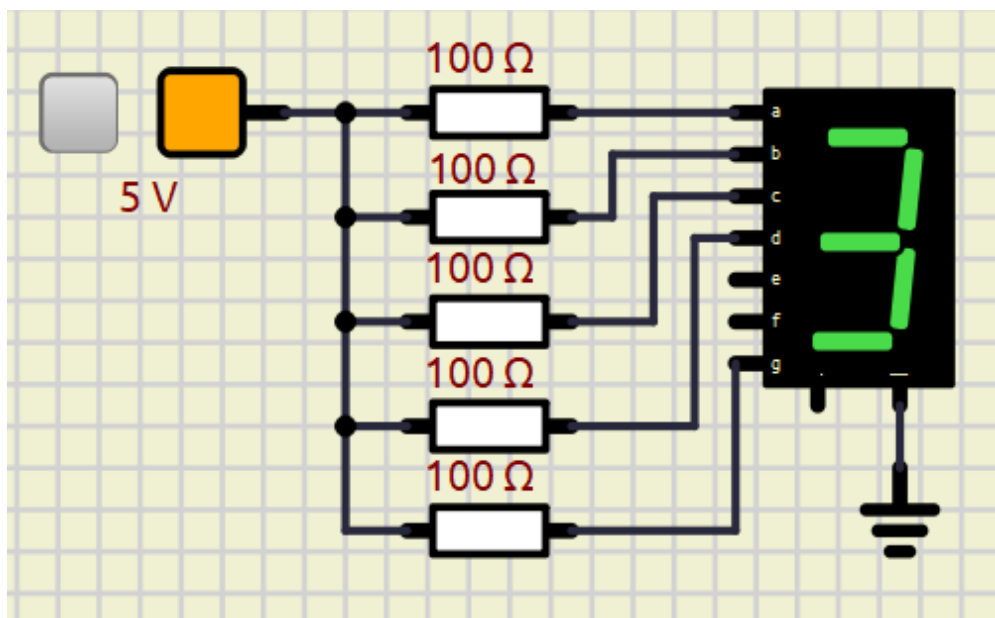




O guia "Electric" mostra os parâmetros elétricos do componente, conforme a cor selecionada para os segmentos (neste caso, Green):



A figura a seguir mostra um arranjo em que é mostrado no display o número 3 em decimal, que corresponde à entrada 0011 em binário (em código BCD), que deve acender os segmentos a, b, c, d, g.

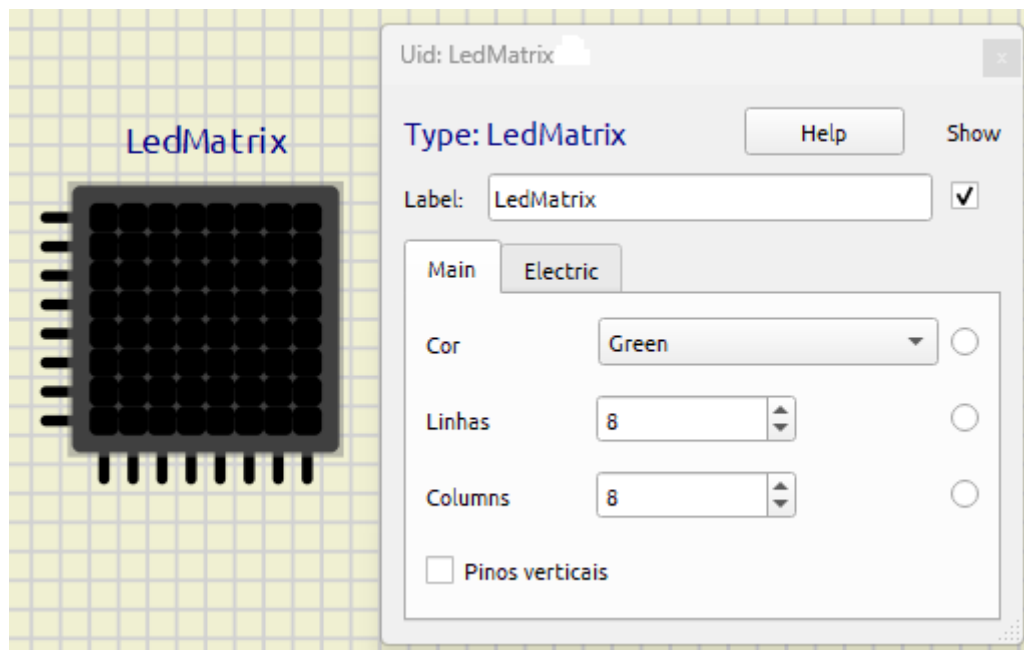


Observa-se que o tipo de display é catodo comum, onde os terminais a, b, c, d, f, são ligados à fonte de alimentação de 5V e o terminal "-" é ligado ao terra.

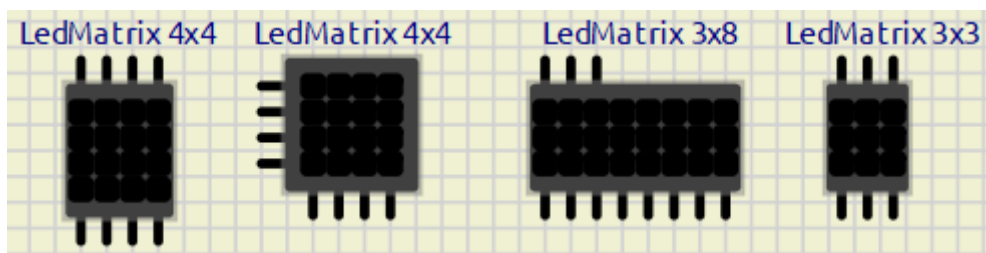
Isto caracteriza um display de catodo comum.

Matriz de LED (LedMatrix):

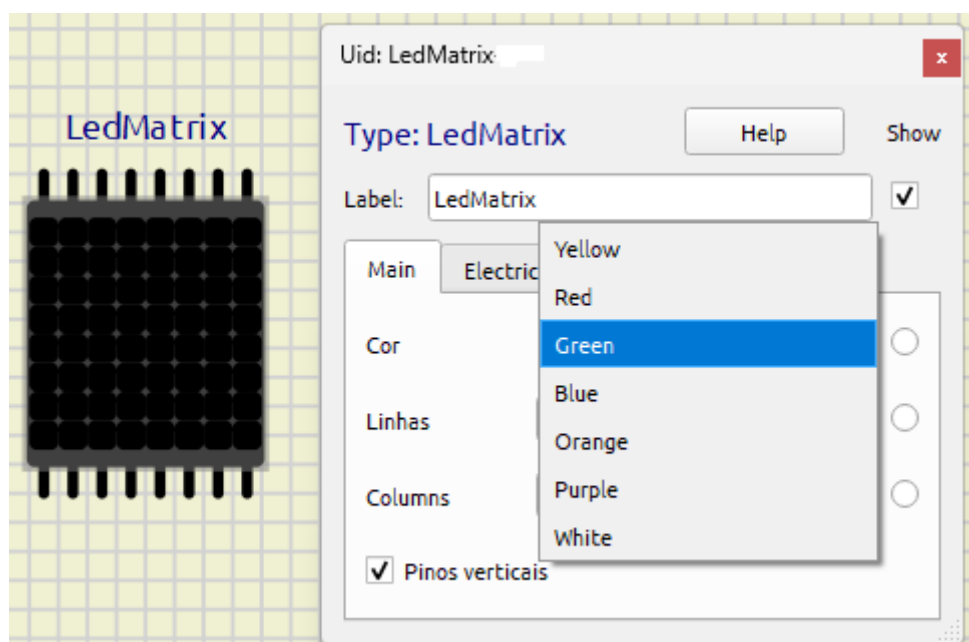
Por padrão, o simulIDE insere o layout mostrado abaixo, de uma matriz de LED, onde temos os pinos que representam as linhas (ROWS) e as colunas (COLUMNS).

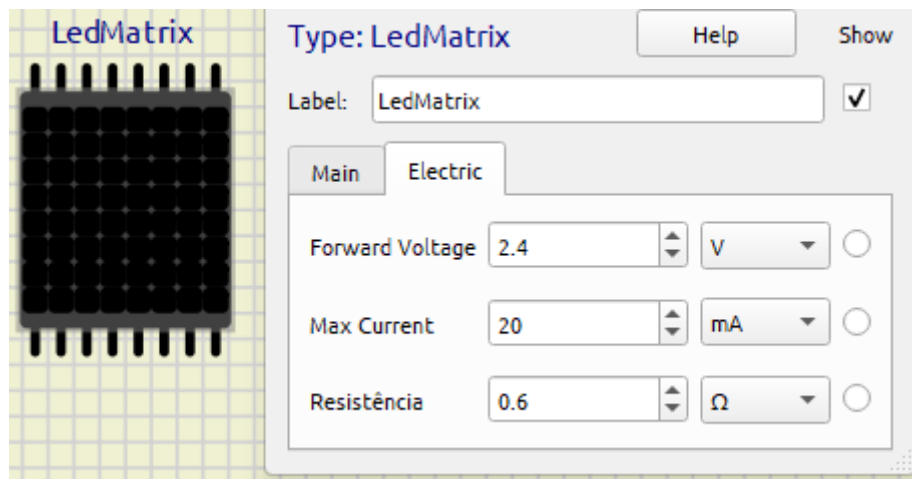


É possível também alterar o formato geométrico da matriz e sua quantidade de pontos nas opções "Linhas" e "Columns":

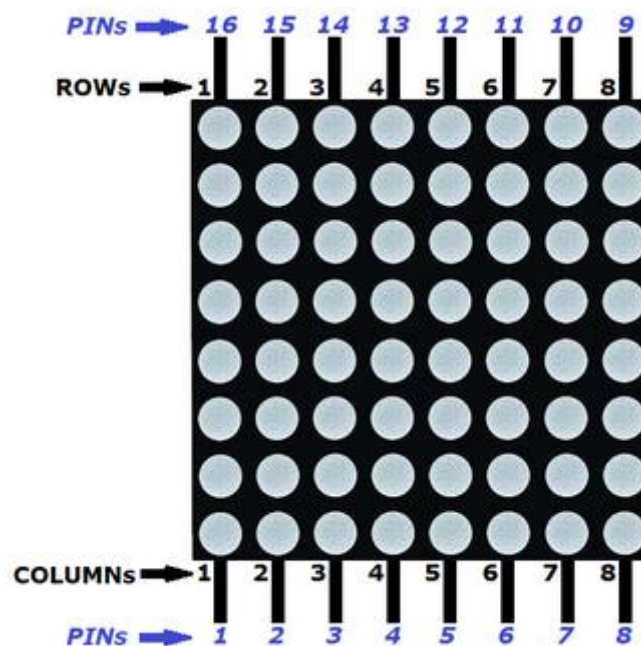


A matriz de LEDs apresentada a seguir é 8x8 com 64 pontos (elementos) com pinos dispostos na vertical.



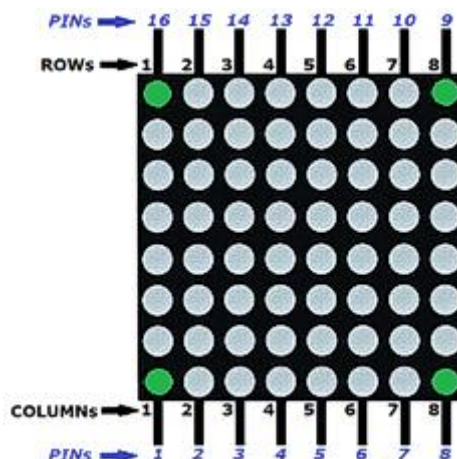


A figura abaixo mostra a configuração de uma matriz disponibilizada no simulIDE em relação a linhas e colunas configuradas no formato 8x8:

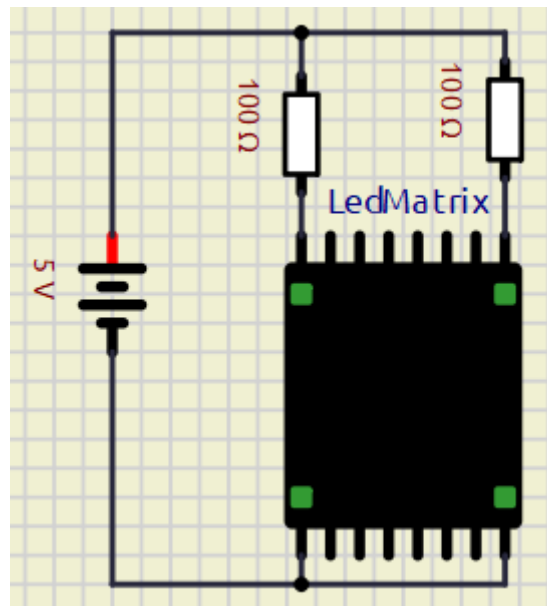


As linhas (ROWS) ligam internamente o anodo (+) e as colunas (COLUMNS) ligam internamente o catodo (-).

Tomemos como um exemplo, como simular uma matriz, de forma que apareçam apenas 4 pontos luminosos nos extremos da tela:



A simulação deverá ser feita conforme diagrama esquemático abaixo, montado no simulador:



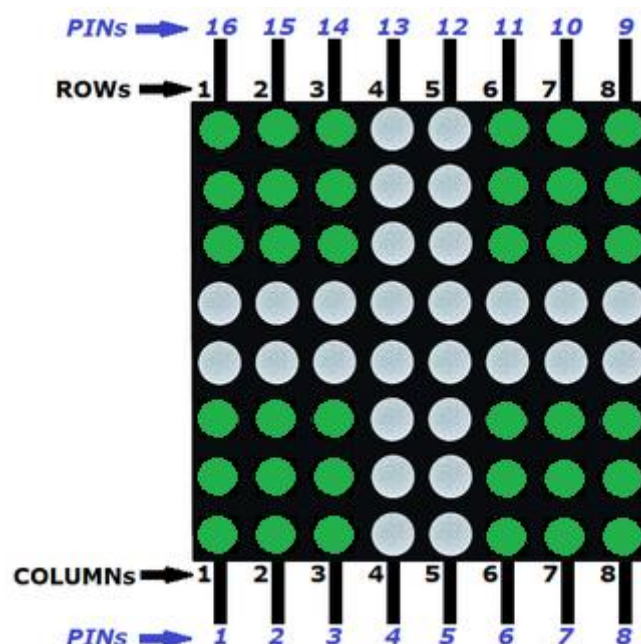
Pinos 1 e 8 → polo (-)

Pinos 9 e 16 → polo (+)

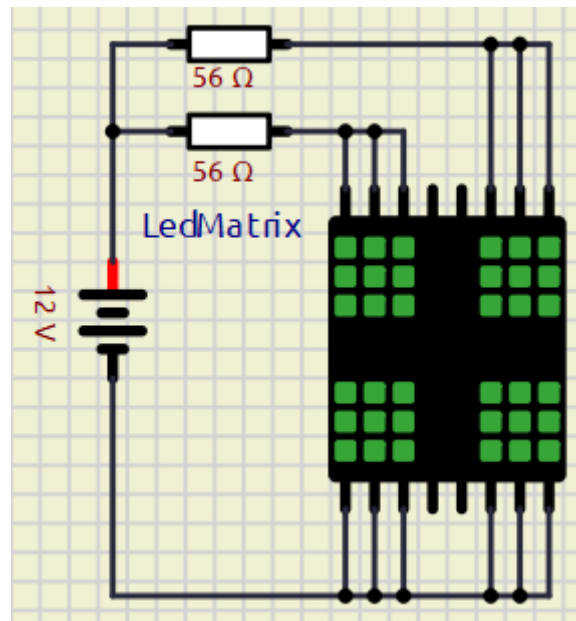
Trata-se de apenas um exemplo para praticar o processo de plotagem, que ocorrem nas matrizes de um modo geral, principalmente quando associadas a microcontroladores.

Uma programação adequada permite visualizar uma grande variedade de símbolos, daí a importância de se conhecer os princípios de plotagem para a implementação de scripts adequados durante a programação.

Vejamos outro exemplo, configurando uma matriz de leds de tal forma a apresentar a figura abaixo ilustrada:



A figura abaixo mostra a montagem e simulação:

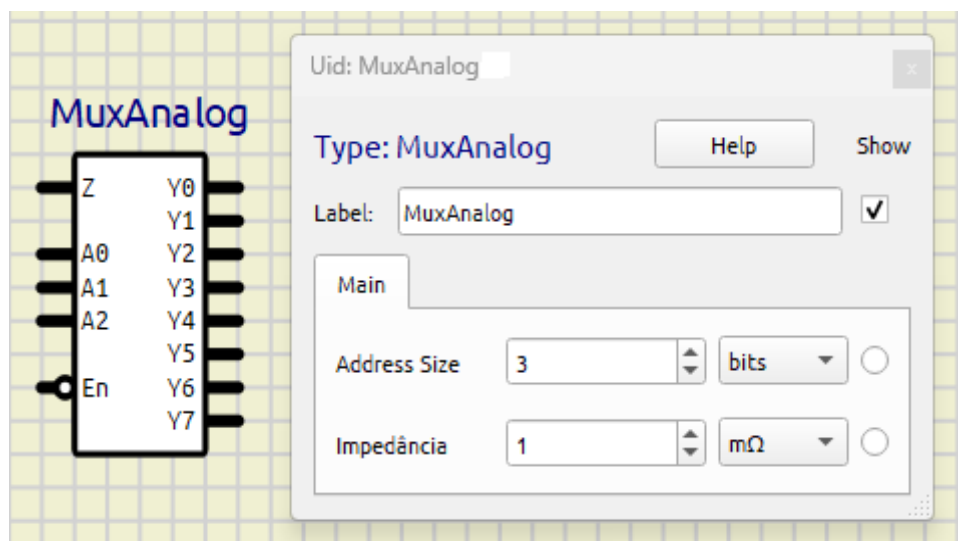


Ativo (Active)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

Multiplexador Analógico (MuxAnalog):

A figura abaixo mostra um Multiplexador Analógico de 3 bits, com endereçamentos de A0 até A2, permitindo que a entrada "Z" seja endereçada a qualquer um dos pinos de saída: Y0 até Y7

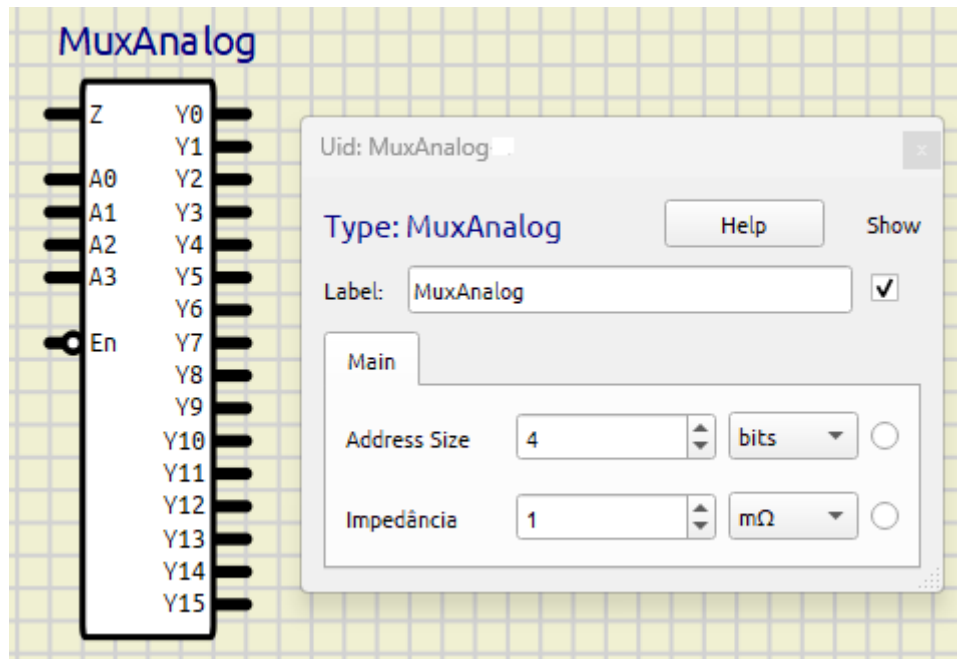


É possível modificar a capacidade do Multiplexador, alterando a quantidade de bits na opção "Address Size". Neste caso, haverá simultaneamente a alteração na quantidade de saídas, lembrando que:

$$\text{Quantidade de saídas} = 2^x$$

Onde "x" é a quantidade de bits da opção "Address Size"

A figura abaixo ilustra um Multiplexador com 4 bits:

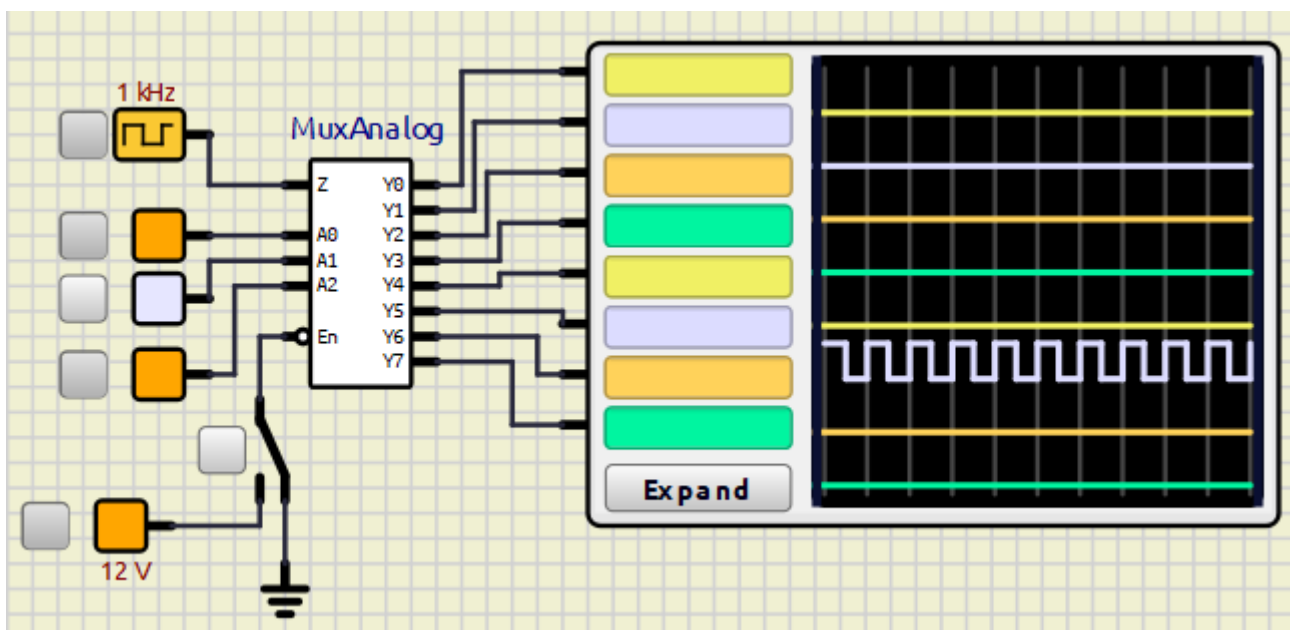


Endereços: A0 até A3

Saídas: Y0 até Y15

A entrada En (Enable) habilita o Multiplexador. Quando aterrada ou nível de referência "0" habilita e quando submetida a uma tensão, desabilita.

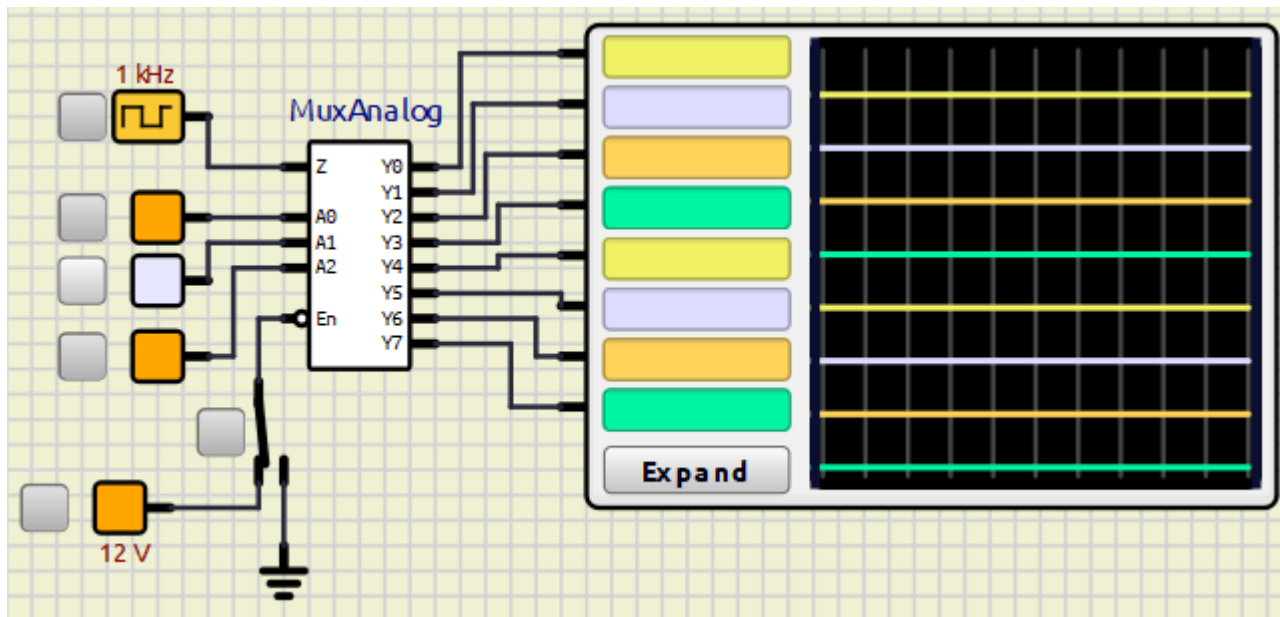
A figura abaixo mostra a montagem do Multiplexador analógico de 3 bits:



O sinal aplicado na entrada "Z" estará presente na saída "Y5", uma vez que o endereçamento A0 = 1, A1 = 0 e A2 = 1, corresponde ao 101 binário (5 decimal).

Observe também que para que isto ocorra, é preciso que o Multiplexador esteja habilitado, e no caso, a entrada "En" está aterrada (tensão - 0V), caso contrário, o

Multiplexador ficará inoperante, conforme mostra a figura a seguir, em que a entrada "En" está submetida a uma tensão de 12V:



Lógicos (Logic) - Portas (Gates)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

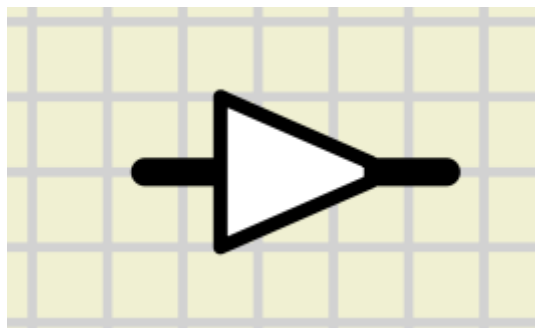
Buffer:

Um "buffer" (neste caso, buffer digital) é um componente que repassa o sinal de entrada para a saída sem alterar o seu nível lógico (0 ou 1).

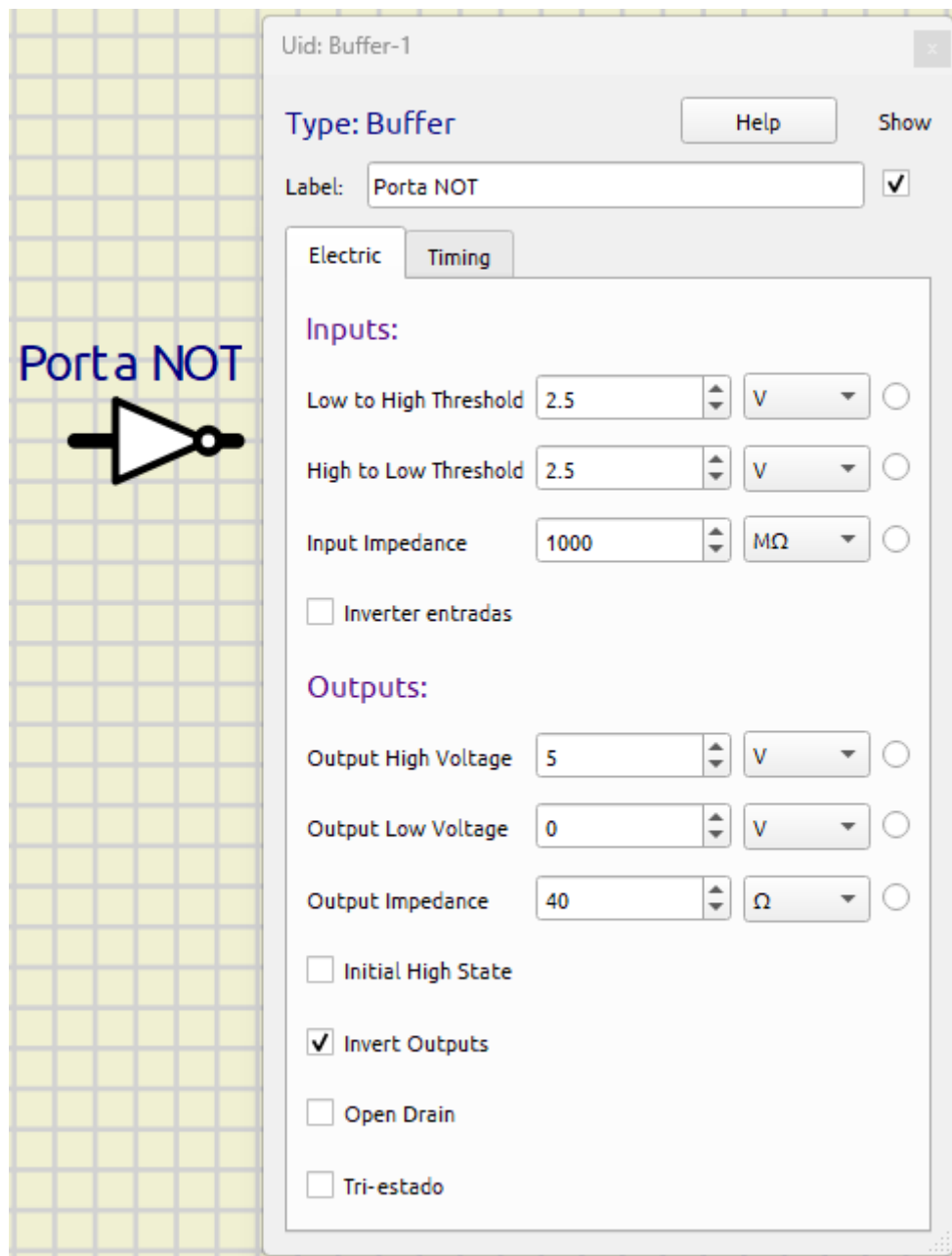
Tem duas funções muito importantes:

a) isolamento do circuito, principalmente no que diz respeito ao isolamento de ruídos;

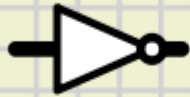
b) reforço, ou seja, aumenta a capacidade de corrente para alimentar outros componentes atuando assim, para a estabilização de um sinal.



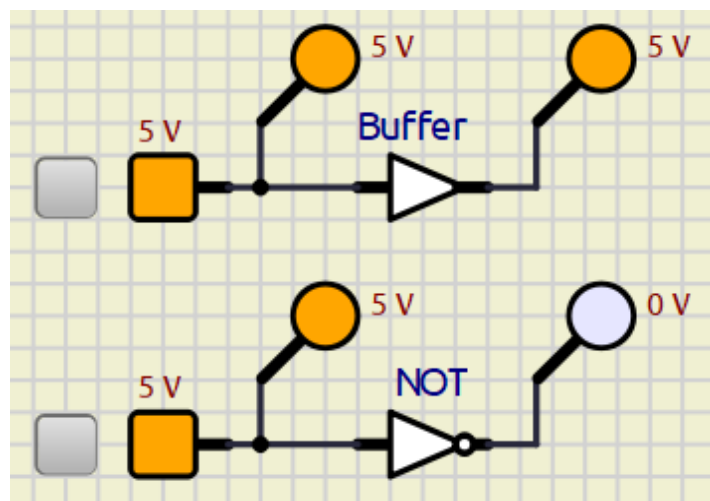
No Simulador IDE, o buffer pode ser configurado para atuar também como uma porta inversora (porta NOT), bastando para isso, na opção "Electric" selecionar "Invert Outputs", conforme ilustra a figura a seguir:



Porta NOT

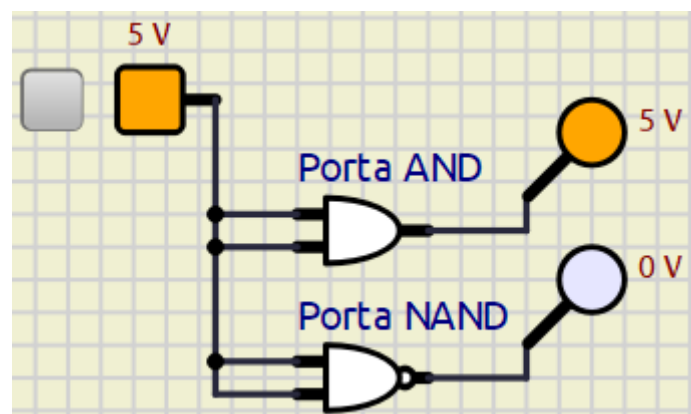
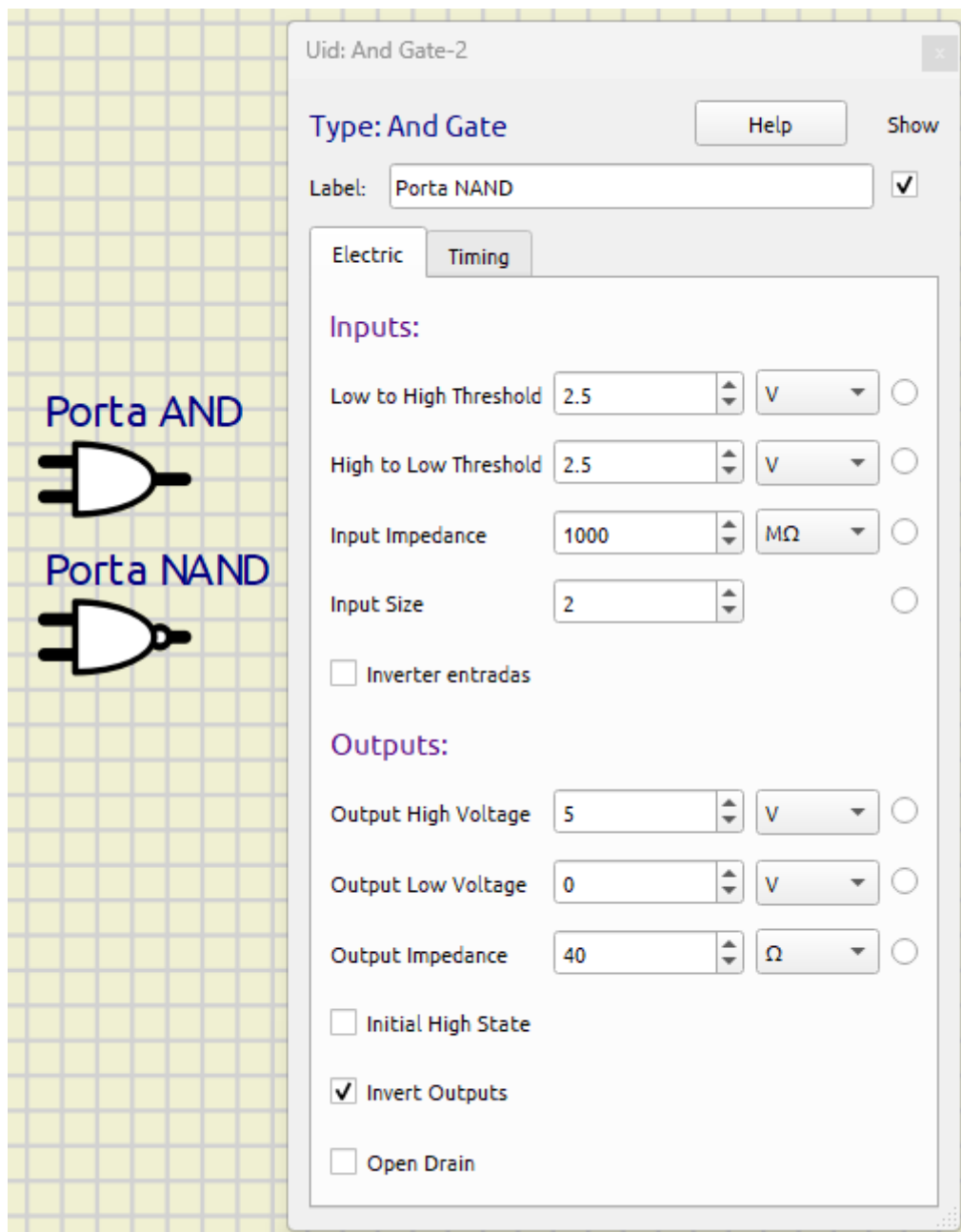


A figura abaixo ilustra uma simulação, comparando um dispositivo “buffer” e uma porta NOT:

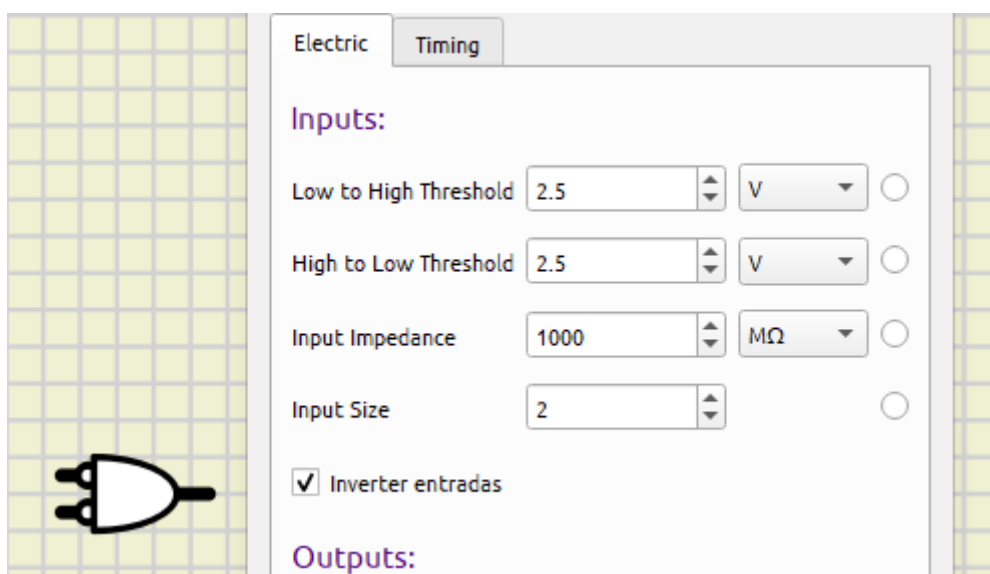


Porta AND (AND Gate):

No Simulador IDE, a porta AND pode ser configurada para atuar também como uma porta NAND, bastando para isso, na opção "Electric" selecionar "Invert Outputs", conforme ilustra a figura abaixo:

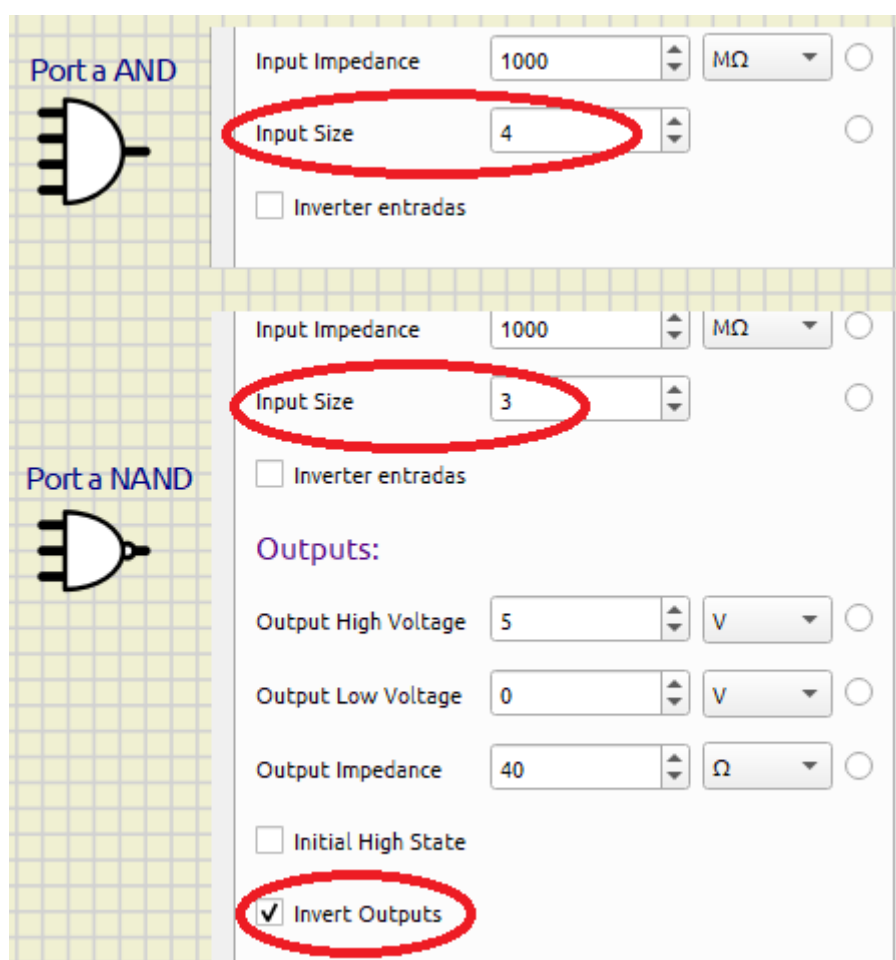


Existe também a possibilidade de inverter as entradas, tanto para a opção porta AND como porta NAND, bastando para isso, na opção "Electric" selecionar "Invert Inputs" ou "Inverter entradas"

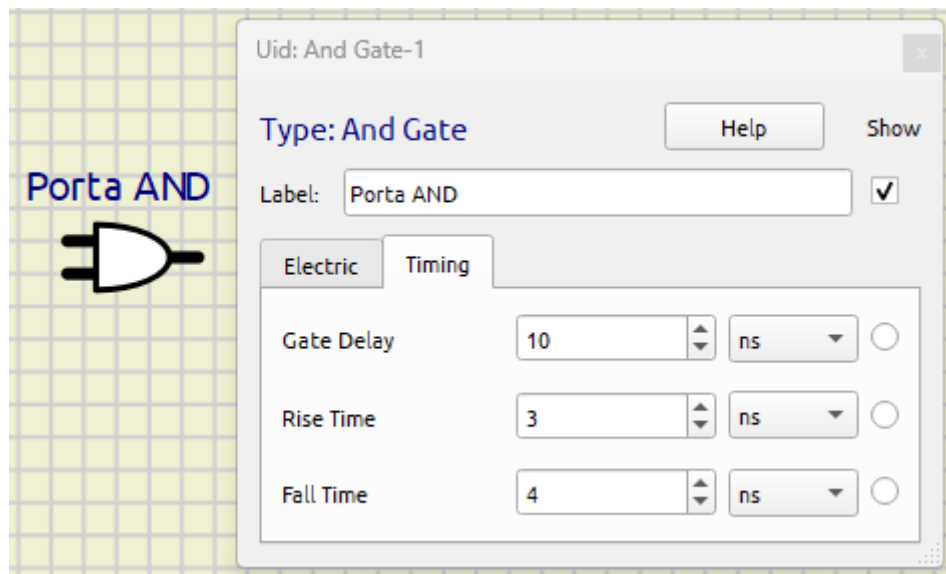


Alguns projetos necessitam de portas lógicas com mais de duas entradas (isto vale para qualquer função lógica).

Na opção "Electric" em "Input Size" podemos definir a quantidade de entradas de uma função lógica. A figura abaixo mostra as funções AND e NAND com 4 e 3 entradas respectivamente:



A opção "Timing" refere-se ao tempo de subida (Rise Time) e tempo de descida (Fall Time), parâmetros estes, que são obtidos através do manual dos fabricantes, e varia de acordo com a tecnologia aplicada na fabricação dos chips como TTL, CMOS, etc.



Porta OR (OR Gate):

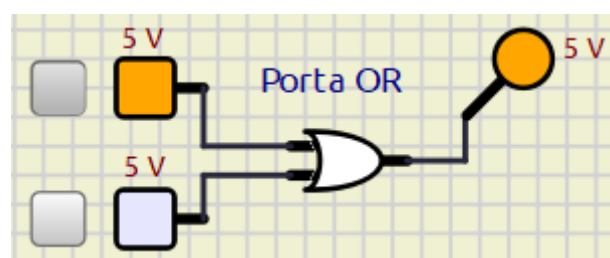
Uma porta OR, no Simulador IDE pode ser configurada também da mesma forma como vimos anteriormente:

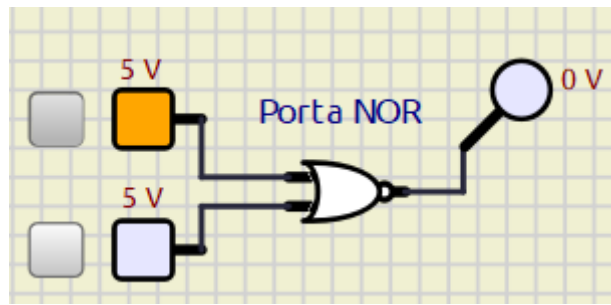
- a) conversão para porta NOR
- b) definição da quantidade de entradas
- c) inversão das entradas
- d) parâmetros da opção "Timing"

Veja abaixo uma porta OR, configurada para 3 entradas:

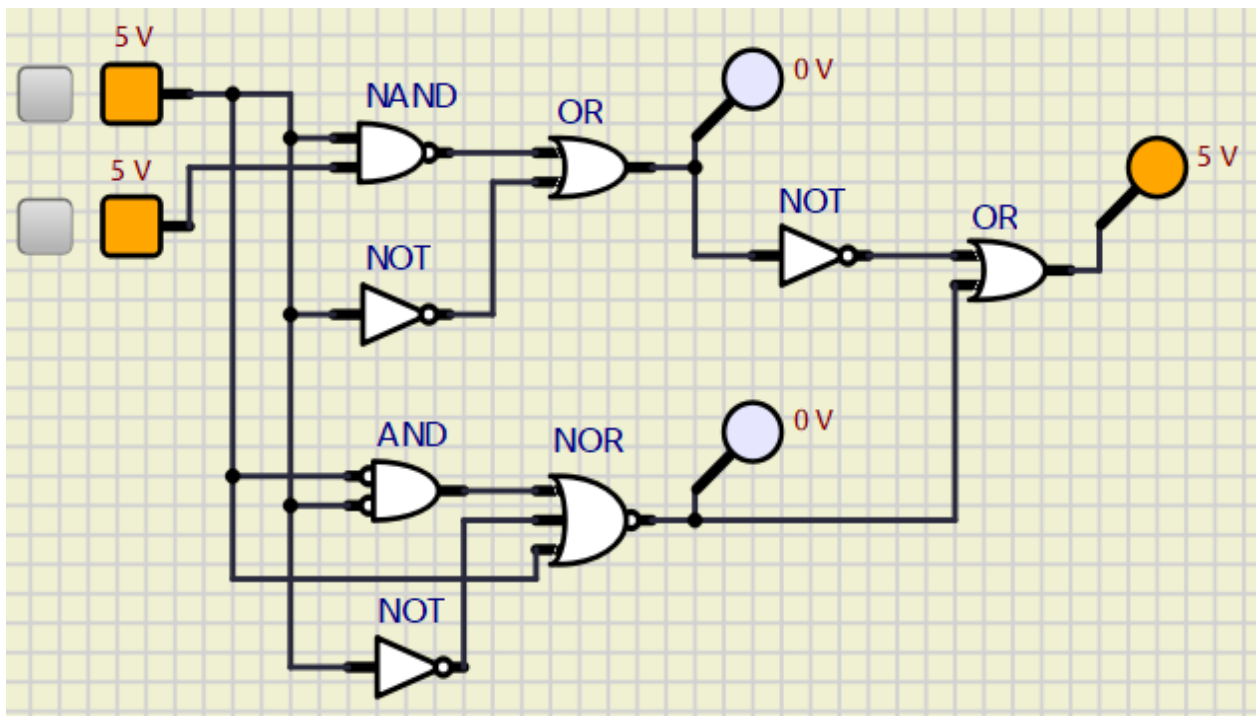


As figuras a seguir representam uma função (porta) OR e NOR respectivamente, com 2 entradas:





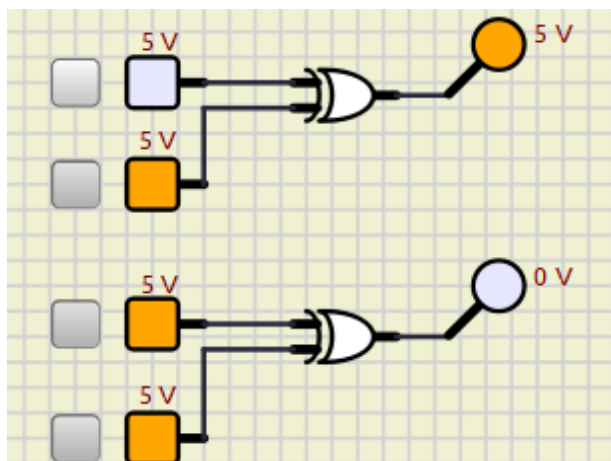
A figura abaixo ilustra uma simulação aplicando as funções lógicas vistas até o momento:



Porta XOR (XOR Gate):

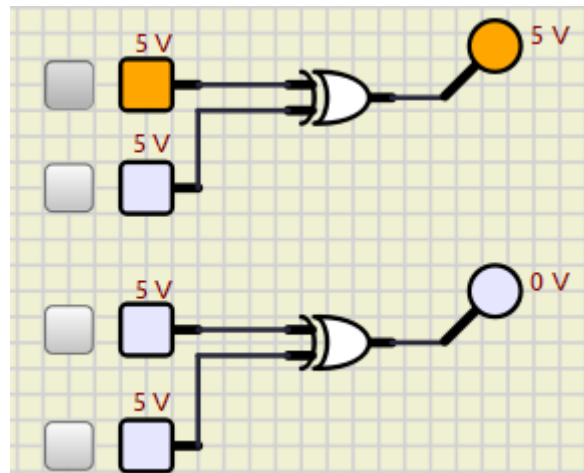
A porta XOR é uma porta lógica que compara bits na entrada. Essa porta também é conhecida como OU EXCLUSIVO.

Se as entradas com nível lógico 1 forem em número ímpar, a saída lógica será igual a 1, se forem em número par a saída lógica será igual a 0. A figura abaixo mostra uma função ou porta XOR, com duas entradas.



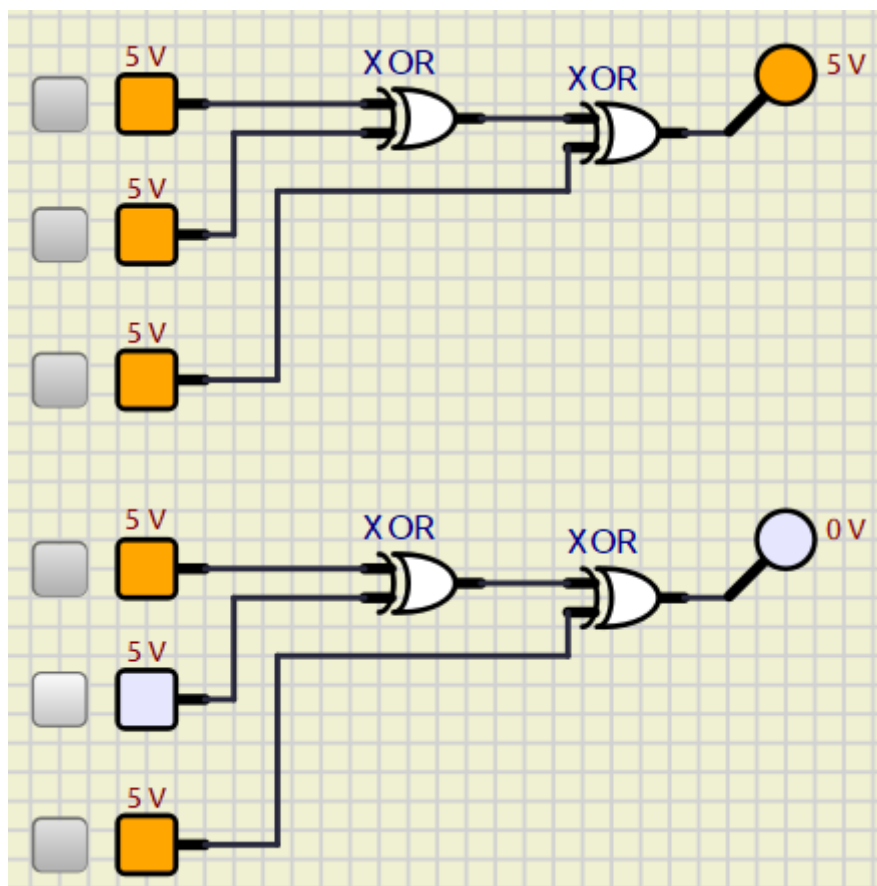
Quando as entradas forem diferentes entre si, a saída terá nível lógico igual a 1. No caso das entradas com o mesmo nível lógico, a saída será igual a 0.

Resumindo: comparando as simulações, quando as entradas forem iguais a 0 ou igual a 1, a saída será igual a 0.



Observa-se que não é possível aumentar a quantidade de entradas na porta XOR ("Input Size"), a exemplo das portas ou funções vistas anteriormente.

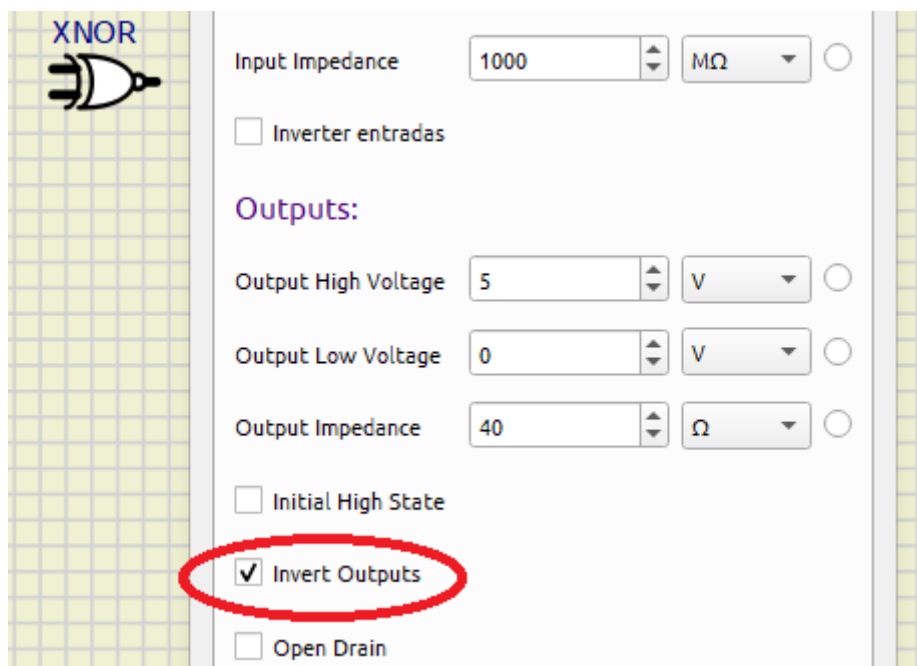
Se quisermos simular uma função XOR com 3 entradas por exemplo, devemos fazer o cascadeamento, conforme mostra a figura abaixo:



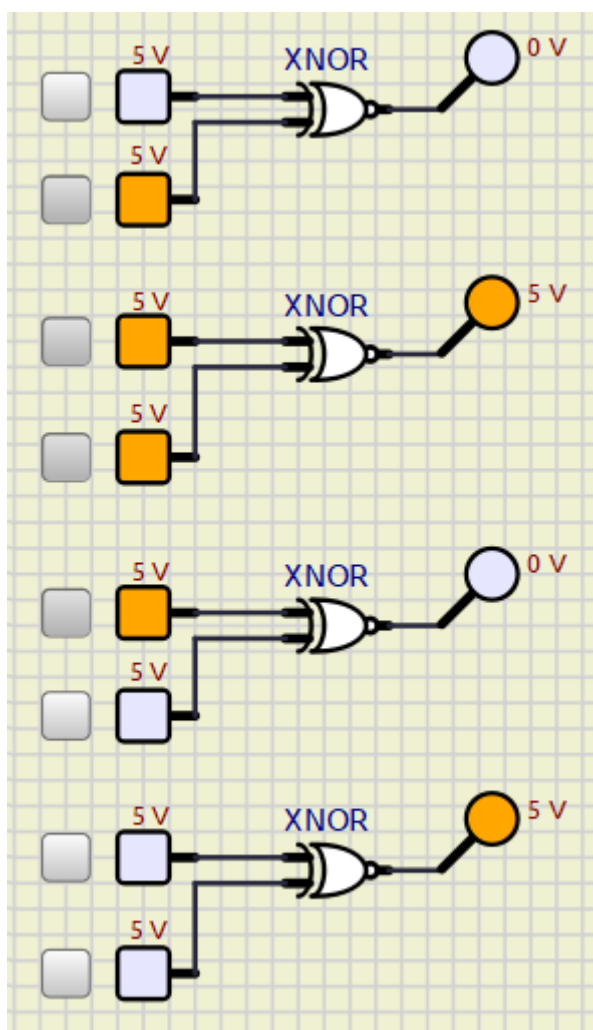
Porta ou função XNOR (Circuito de Coincidência)

Uma porta ou função XNOR é uma função ou porta XOR com um inversor na saída.

Portanto, podemos afirmar que uma porta ou função XNOR é o complemento de uma porta ou função XOR.



Resumindo: a) entradas com nível lógico igual, saída igual a 1; b) entradas com níveis lógicos diferentes, saída igual a 0

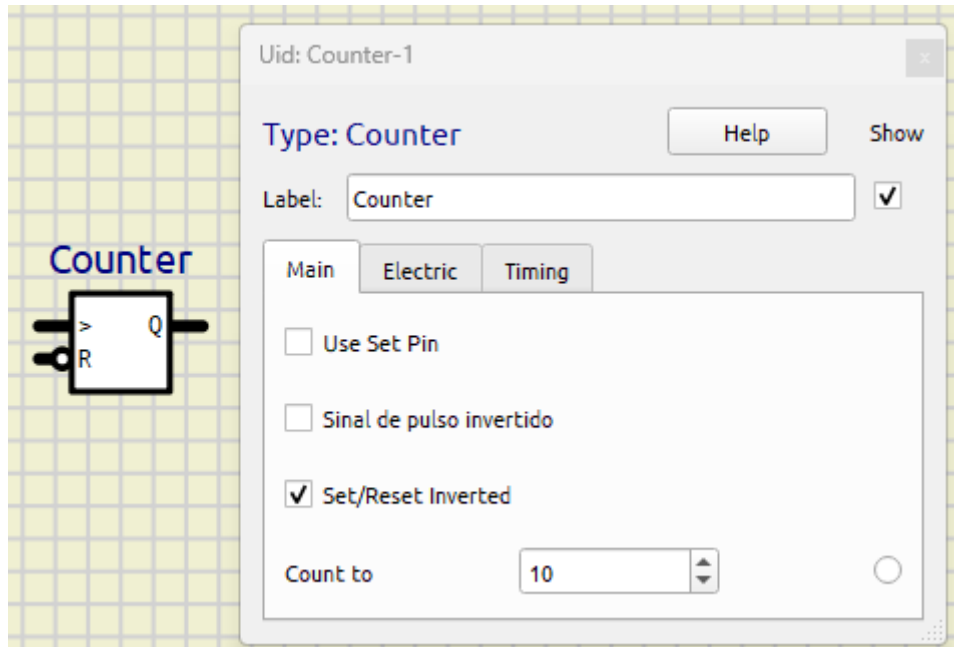


Lógicos (Logic) - Aritméticos (Arithmetic)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

Binary Counter:

Aplicação comum: simulação de temporizadores



Na opção "Count to" ao se aplicar pulsos na entrada depois de uma determinada contagem a saída é acionada.

Na caixa de diálogo acima, a contagem para ativar a saída é 10 pulsos.

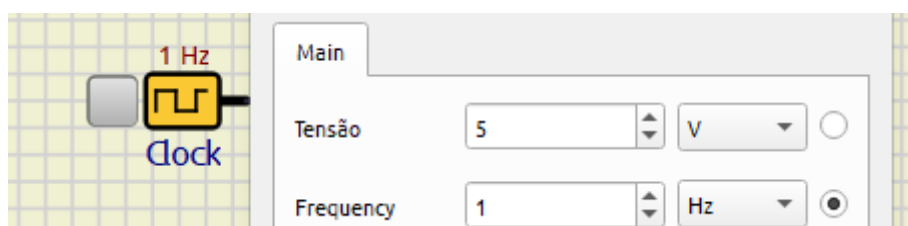
A entrada "R" (Set/Reset) pode ser configurada em 0 ou 1, dependendo das características do projeto.

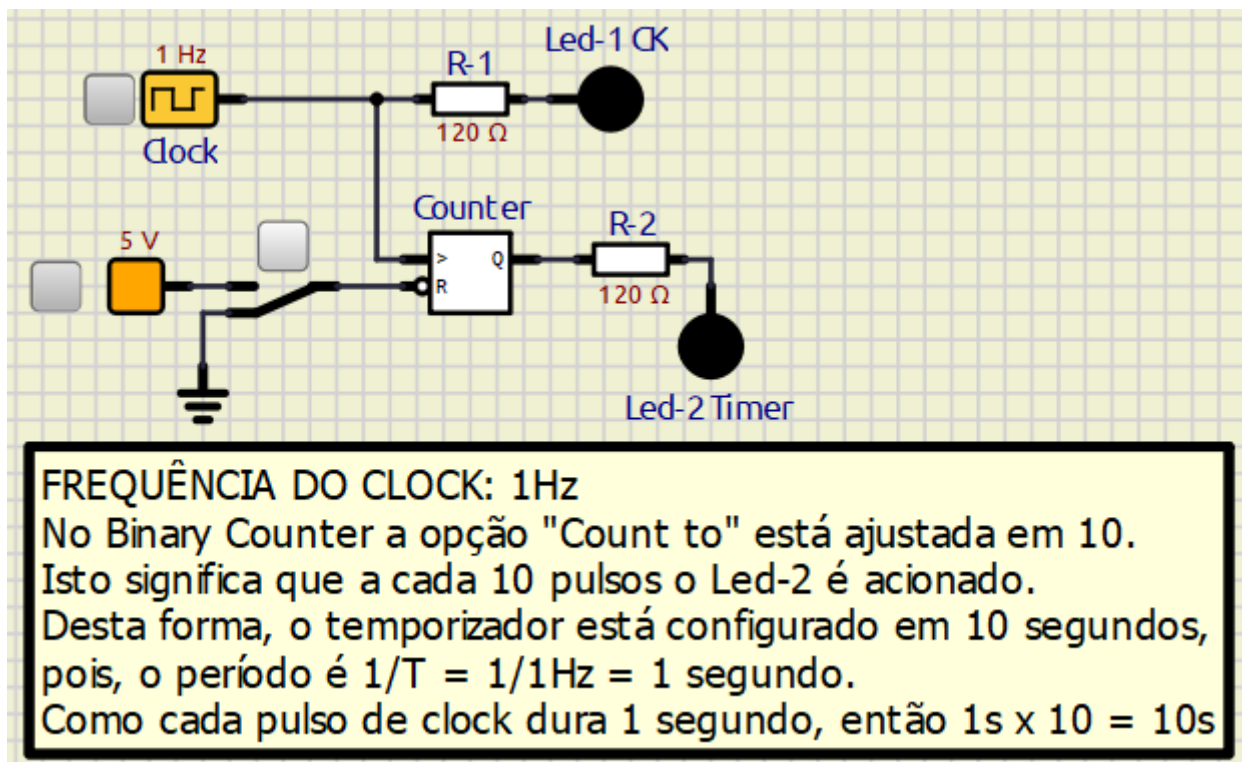
Veja abaixo um exemplo da aplicação de um temporizador "Timer" de 10 segundos.

- 1) a frequência do clock foi ajustada para 1Hz
- 2) como o período é o inverso da frequência, temos então $T = 1/f$

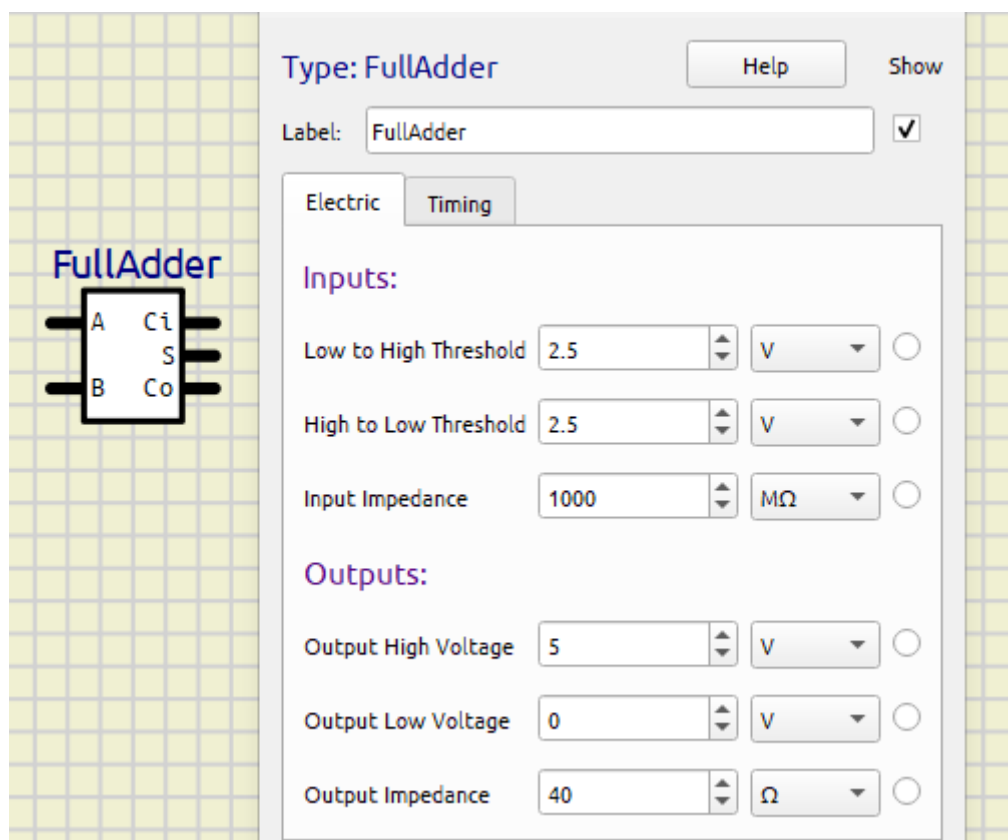
$$T = 1/1\text{Hz} = 1 \text{ segundo}$$

$$10 \text{ pulsos} \times 1 \text{ segundo} = 10 \text{ segundos}$$

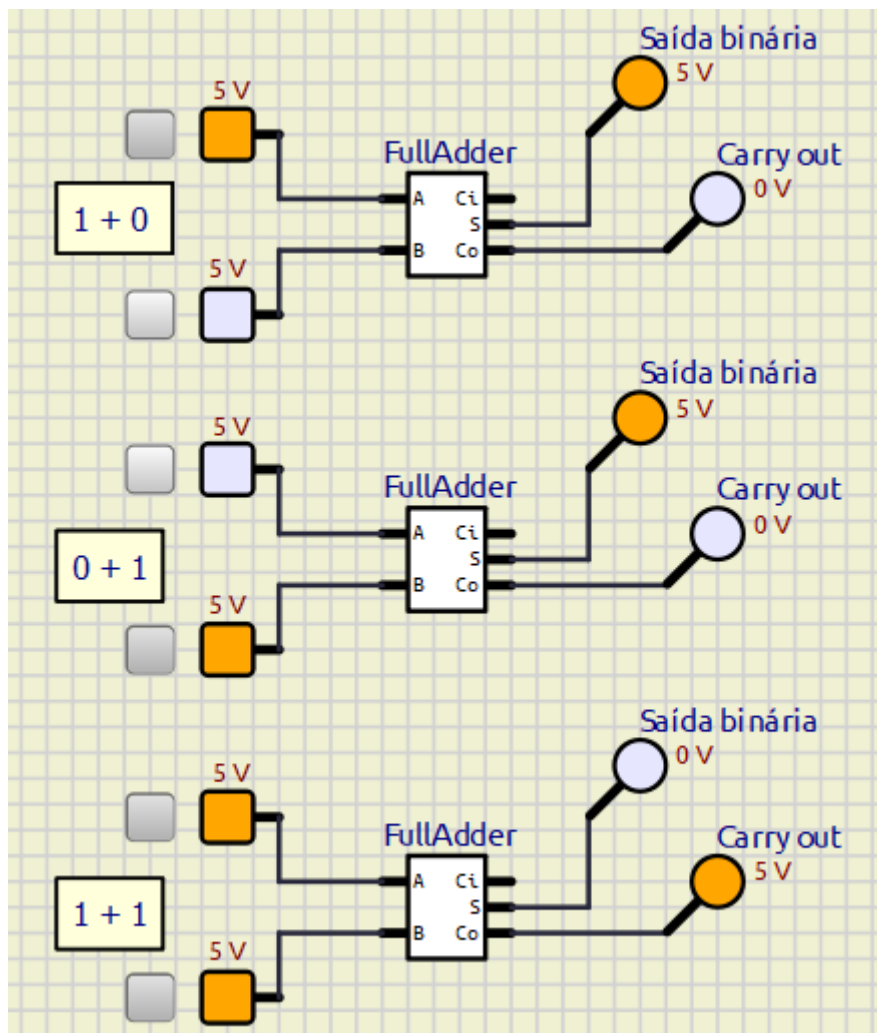




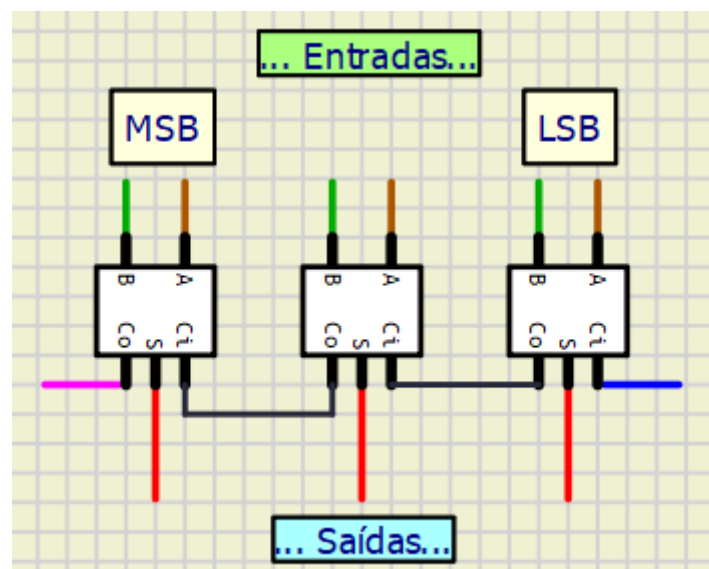
Somador Completo (Full Adder):



A e B - entradas
Ci - Carry in (transporte anterior)
Co - Carry out (saída de transporte)
S - Saída



A figura abaixo ilustra um somador paralelo, uma vez que o transporte obtido a cada soma é conectado à entrada do próximo módulo somador.



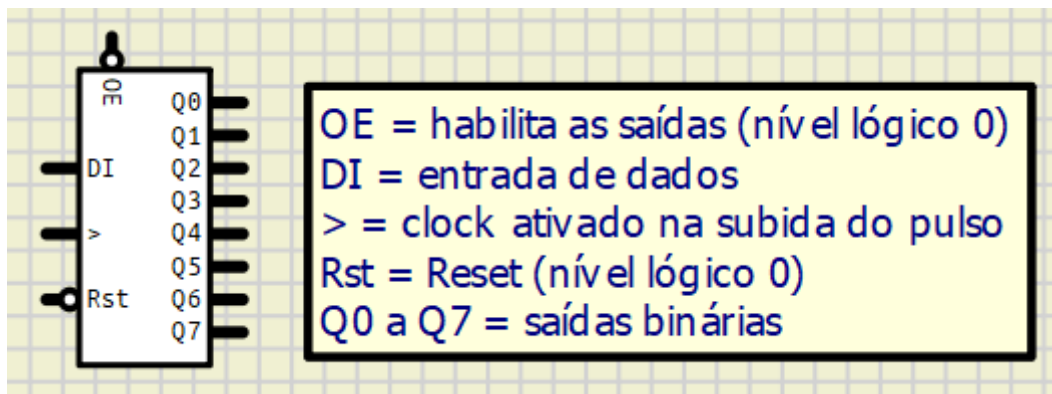
Shift Register:

O registrador de deslocamento (do inglês *Shift-Register*) é um dispositivo largamente usado em sistemas digitais, desde uma simples calculadora de bolso, teclados para introdução de códigos e teclados para computadores.

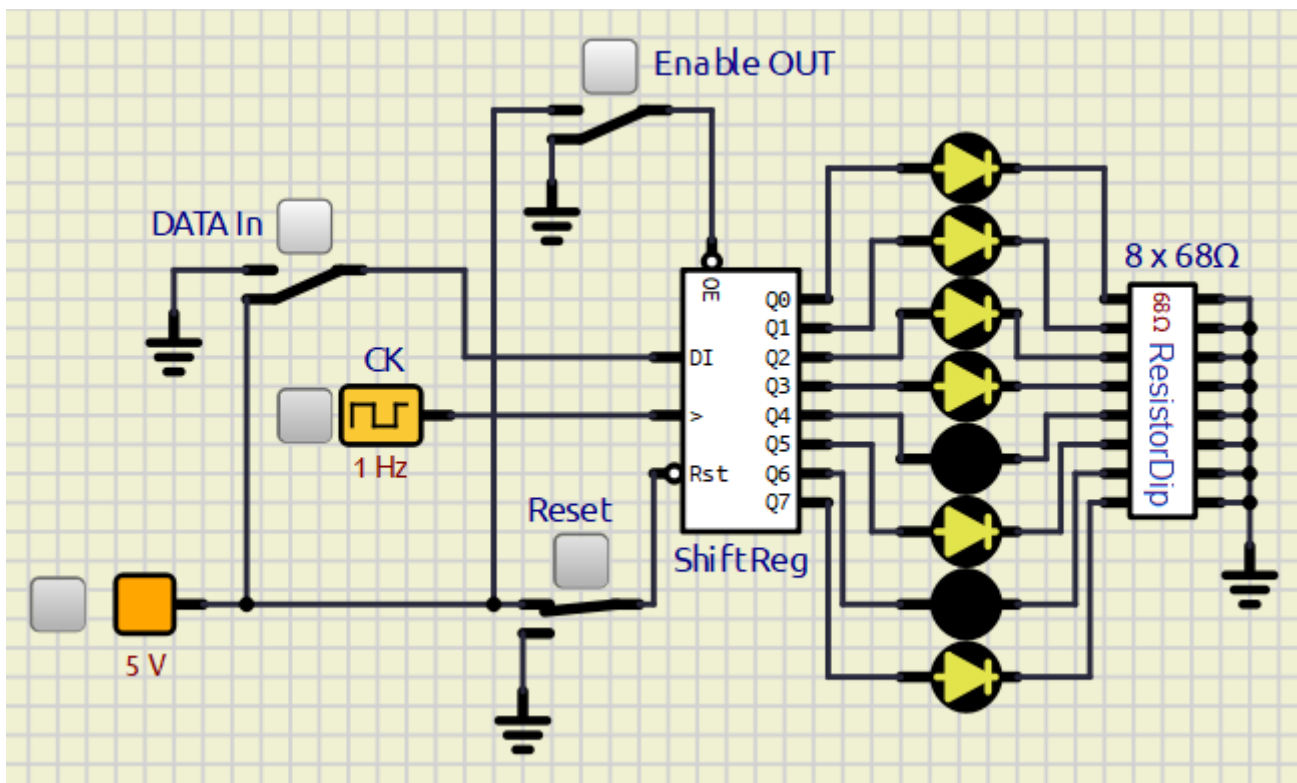
O registrador de deslocamento por sua vez é um dispositivo síncrono no qual os dados ou informações podem ter entrada sequencial (serial) ou paralela, permanecendo até a sua saída que poderá ser também sequencial (serial) ou paralela.

Resumindo, um registrador de deslocamento pode ter uma entrada serial ou paralela e uma saída também serial ou paralela.

O registrador disponível no SimulIDE é do tipo SIPO (Serial In / Parallel Out), ou seja, os dados entram em série e ficam disponíveis em paralelo.



A figura abaixo, mostra uma simulação clássica para o Shift Register SIPO:



O chip "Resistor DIP" tem a finalidade de limitar a corrente em cada uma das saídas.

A escolha do chip (ResistorDip) é opcional, podendo cada uma das saídas serem interligadas a um resistor de 68Ω e referenciados ao terra.

Função (Function):

É um componente muito interessante, pois permite executar operações lógicas sem a utilização de portas lógicas primárias, ou seja, através de programação.

Funções que podem ser configuradas:

Logic operations (Operações lógicas):

Not: ! ➔ ! i0 ou ! i1

And: &

Or: |

Xor: ^

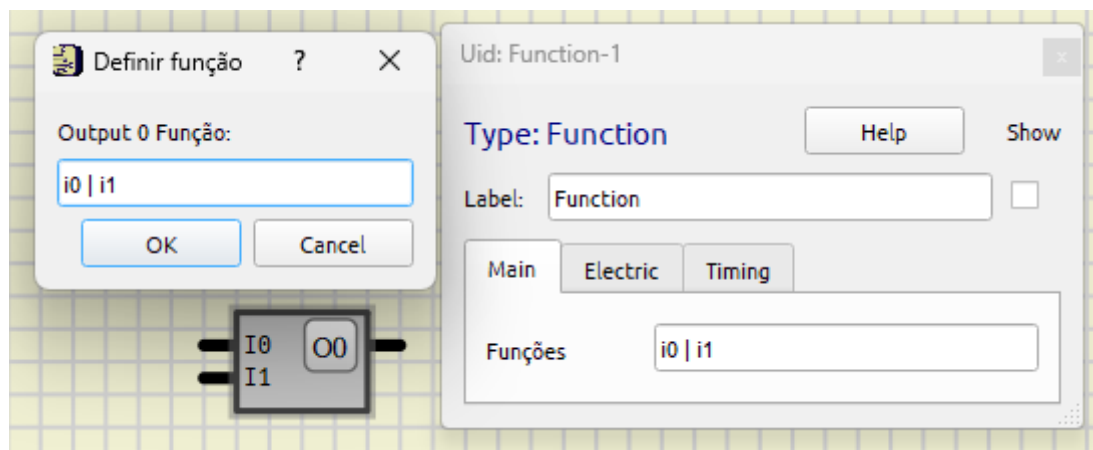
Logic states (Estados lógicos):

- Input n: "in"
- Output n: "on"

A figura abaixo ilustra uma função OR, com 2 entradas (I0 e I1), através da programação na opção "Funções".

A inserção do componente com duas entradas e função OR, é padrão do simulador, cuja configuração pode ser modificada, de acordo com as necessidades do projeto.

Dentro da área de trabalho para fins de simulação, podem ser criados vários blocos, com variadas operações lógicas, a exemplo do que é possível simular utilizando portas lógicas primárias.



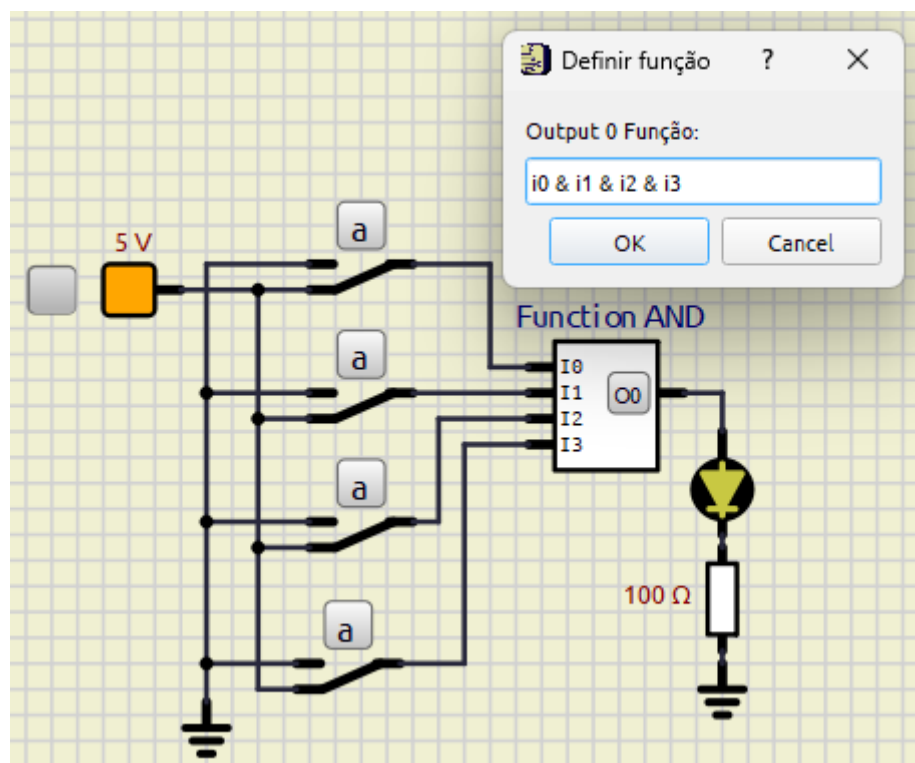
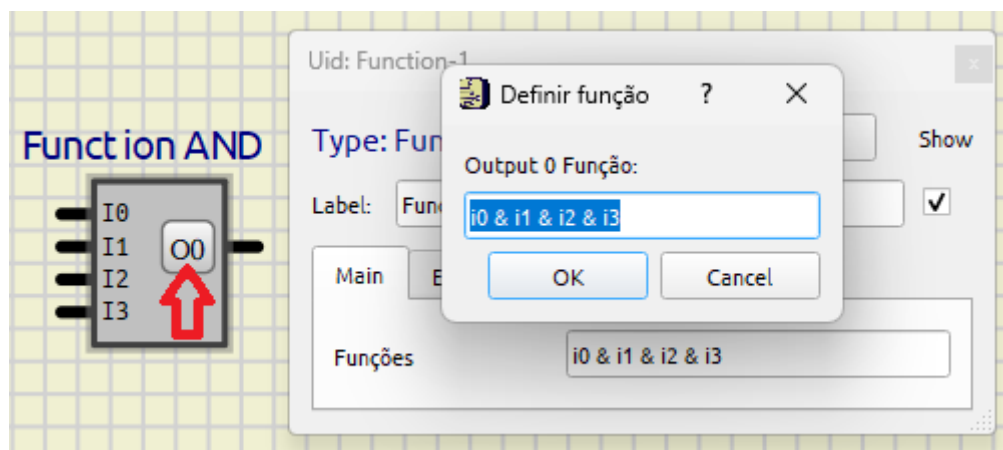
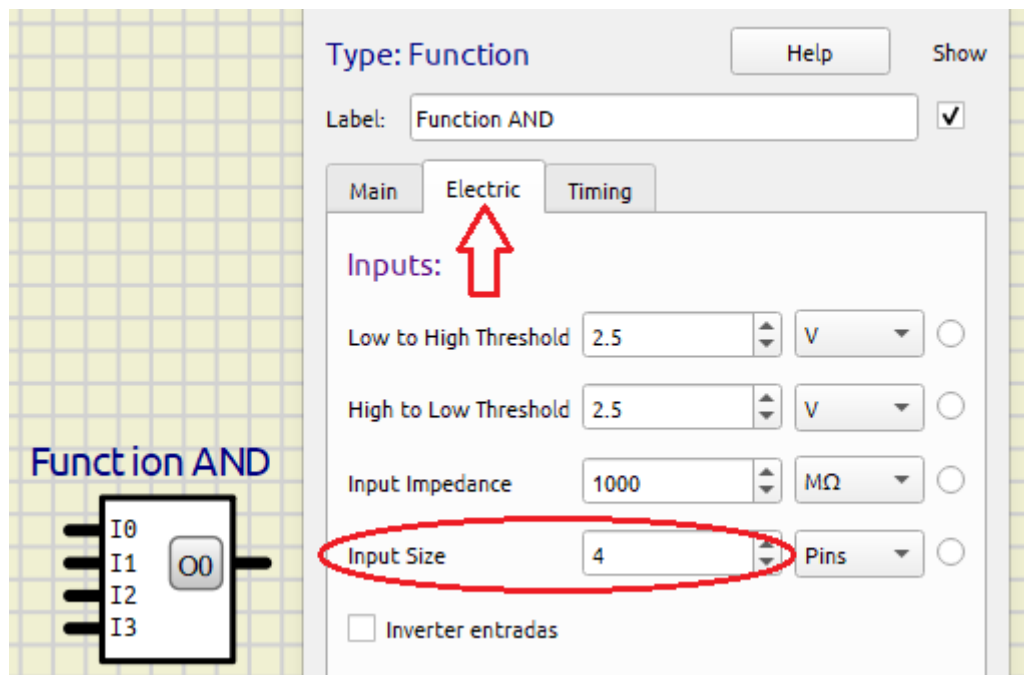
Vamos criar uma função do tipo AND, com 4 entradas.

a) Primeiro passo: na opção "Electric" ajustar "Input Size" para 4. Caso o projeto exija, as entradas podem ser invertidas marcando a opção "Inverter entradas".

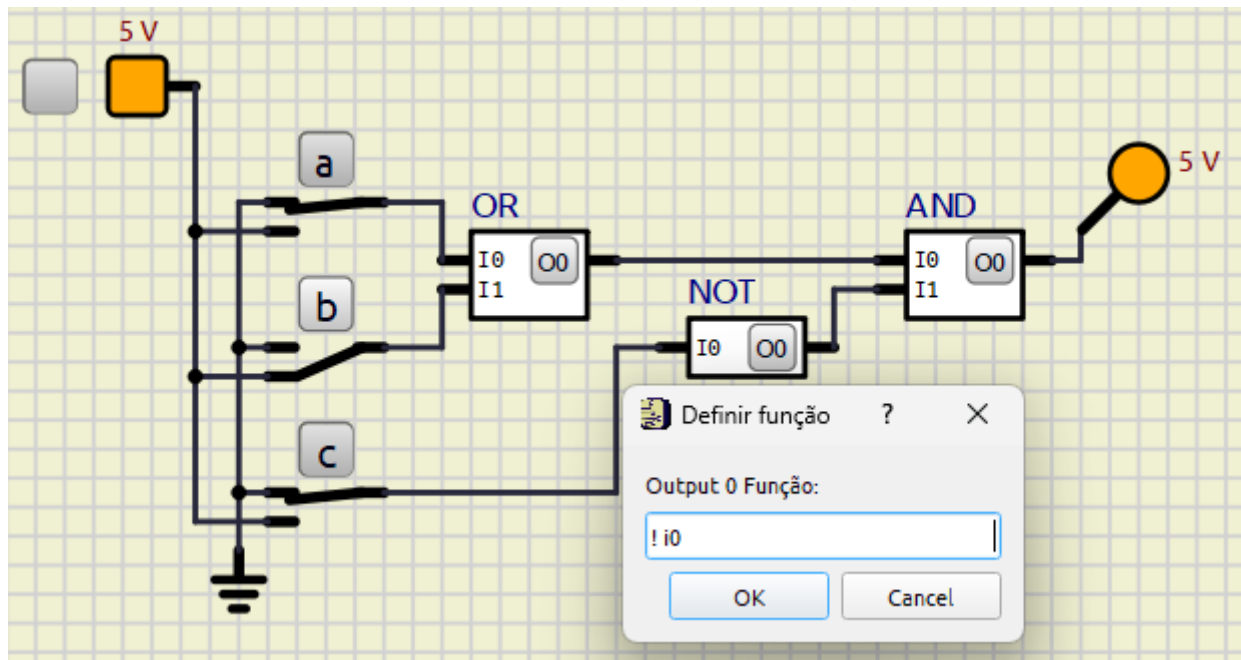
b) Segundo passo: escrever a equação correspondente a operação lógica AND, na opção "Main" no campo "Funções".

i1 & i2 & i3 & i4

Veja a orientação nas figuras a seguir:



A figura abaixo ilustra um exemplo de simulação com as funções (operações lógicas) OR, NOT e AND

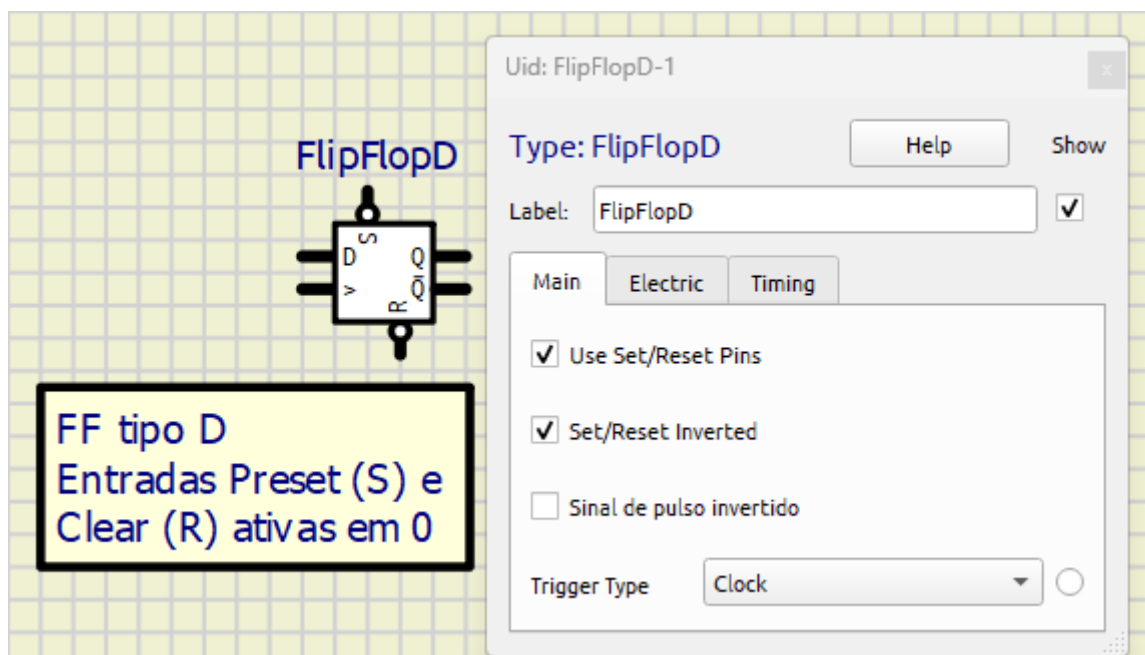


Lógicos (Logic) - Memórias (Memory)

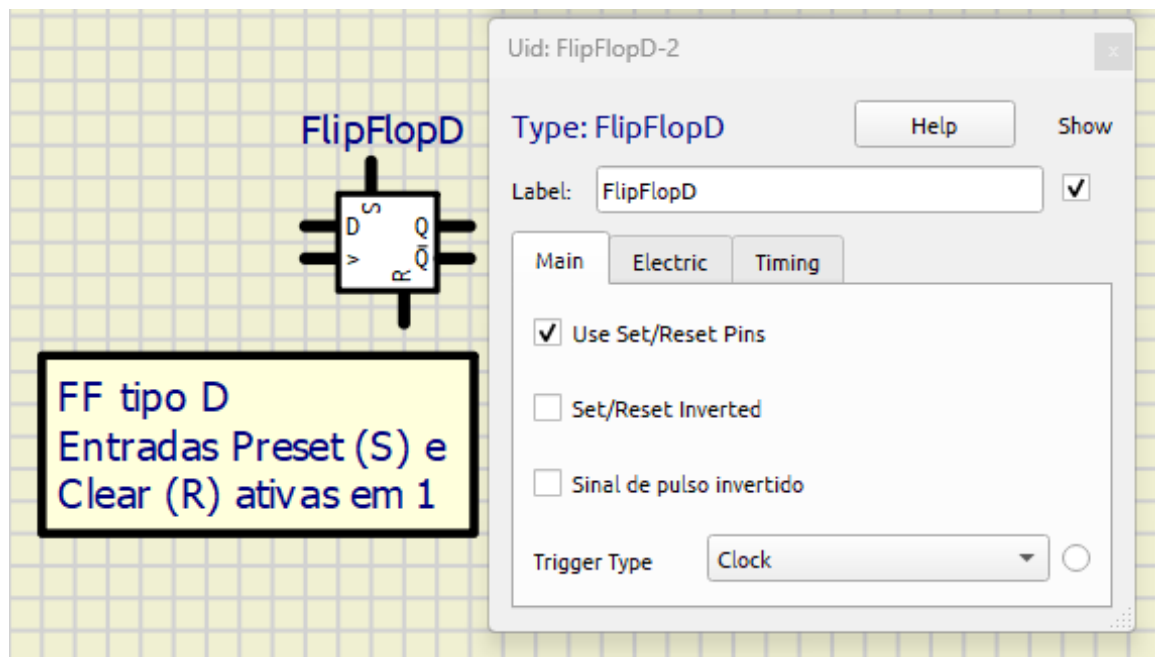
Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou **botão direito do mouse – propriedades.**

Flip Flop D:

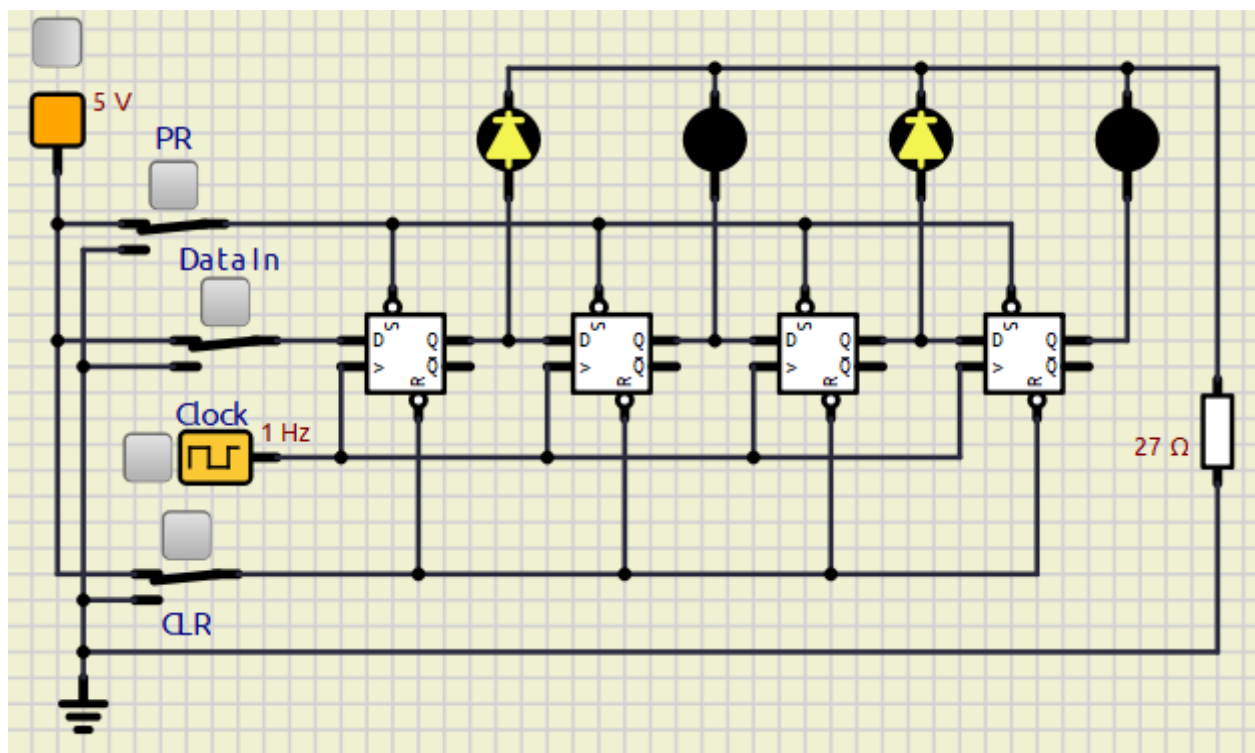
A figura abaixo mostra a apresentação do FF tipo D quando inserido na área de trabalho, onde as entradas assíncronas PR e CLR (Set/Reset) podem ser ativadas também em nível lógico 1.



Ao desmarcar a opção "Set/Reset Inverted" as mesmas serão ativadas em nível lógico 1.



A figura abaixo mostra a construção de um Shift Register SIPO, implementado com FFs tipo D:

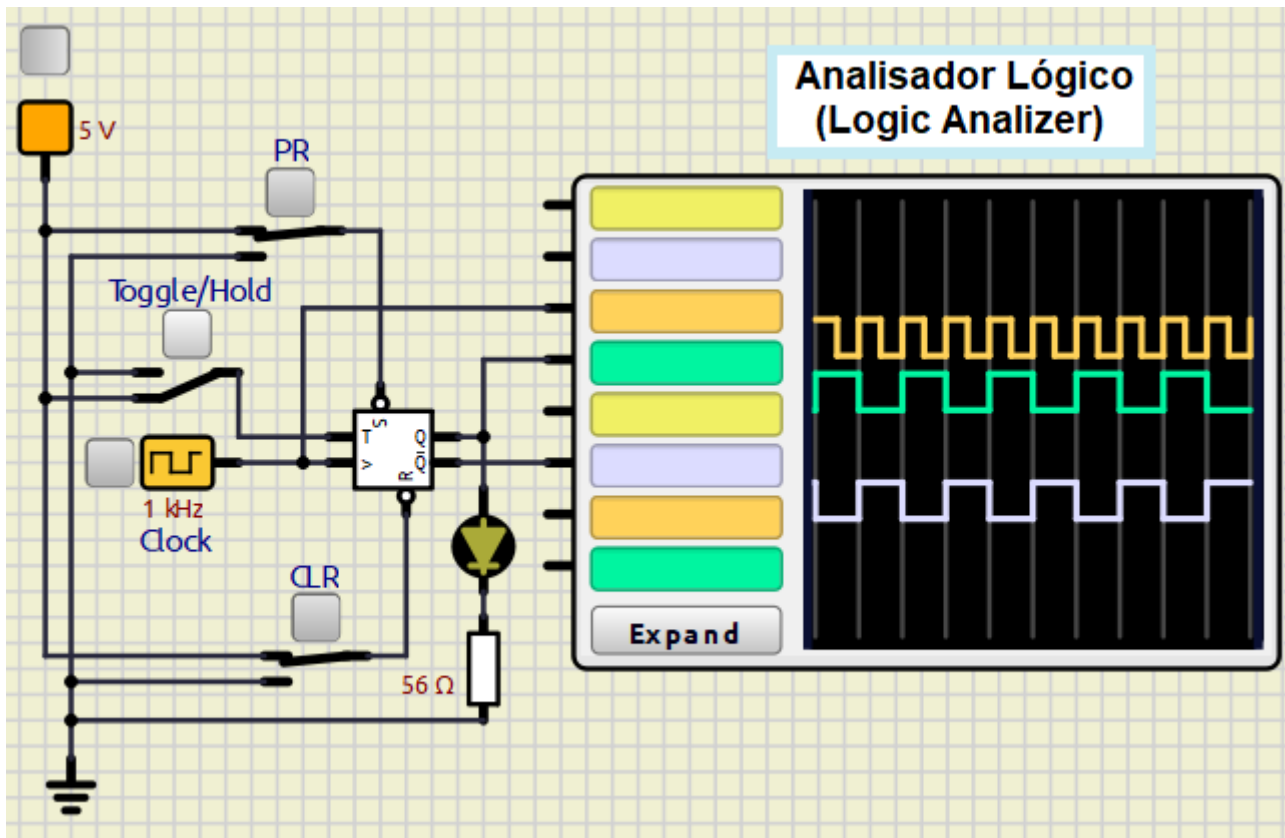


Flip Flop T:

O flip-flop tipo T é um circuito digital biestável que armazena um bit de informação e inverte (comuta) o estado de sua saída a cada pulso de clock quando sua entrada está ativa.

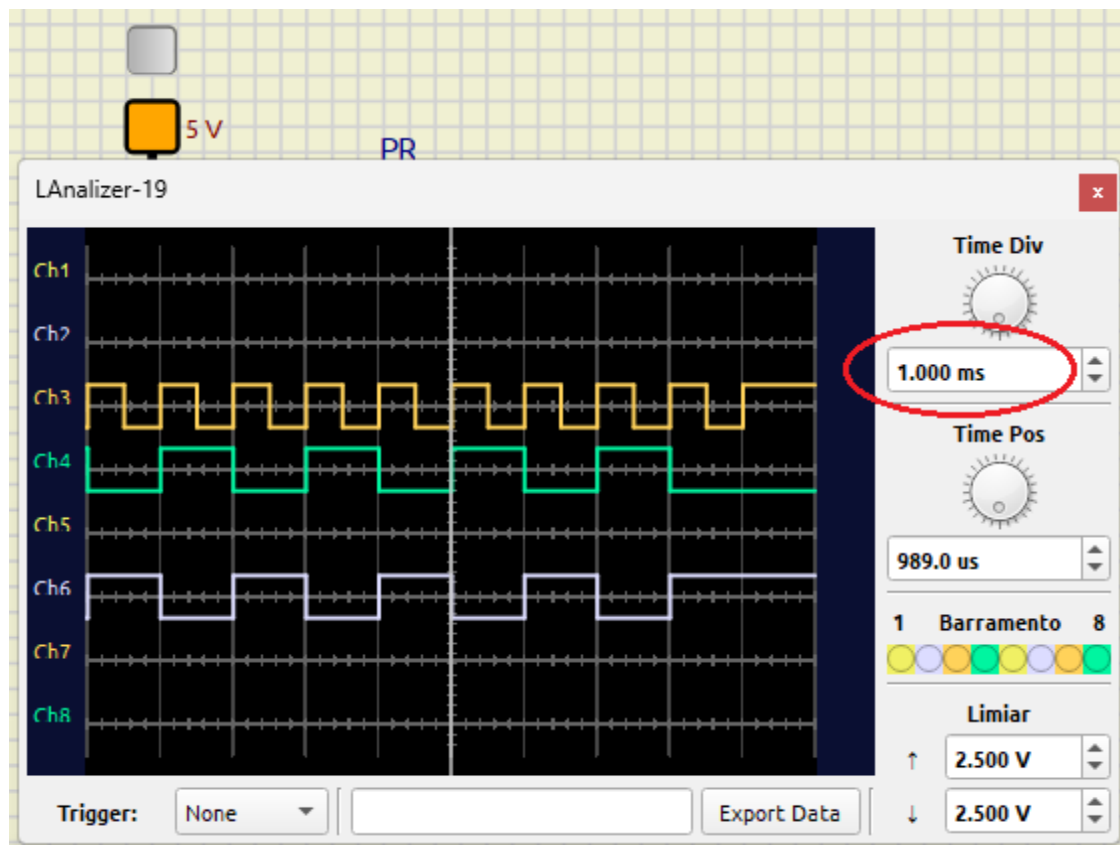
Podemos então dizer que o FF tipo T, possui apenas dois estados: Hold (do inglês: manter) e Toggle (do inglês: alternar ou trocar).

A figura abaixo ilustra o FF tipo T operando como divisor de frequência por 2:



O sinal de clock, com frequência de 1kHz é mostrado na cor laranja (canal 3) e o sinal na saída Q (canal 4, cor verde) está dividindo a frequência pela metade.

Logo abaixo, temos o sinal na saída Q' que está invertido em relação à saída Q.



Flip Flop RS:

O flip-flop tipo RS (também chamado de SR) é um circuito digital biestável usado para armazenar um único bit de informação (0 ou 1).

Ele possui duas entradas principais: R (*Reset* ou limpar) e S (*Set* ou definir).

As saídas Q e Q' se complementam, e a saída Q é o referencial que determina o modo de operação do FF.

Modos de operação:

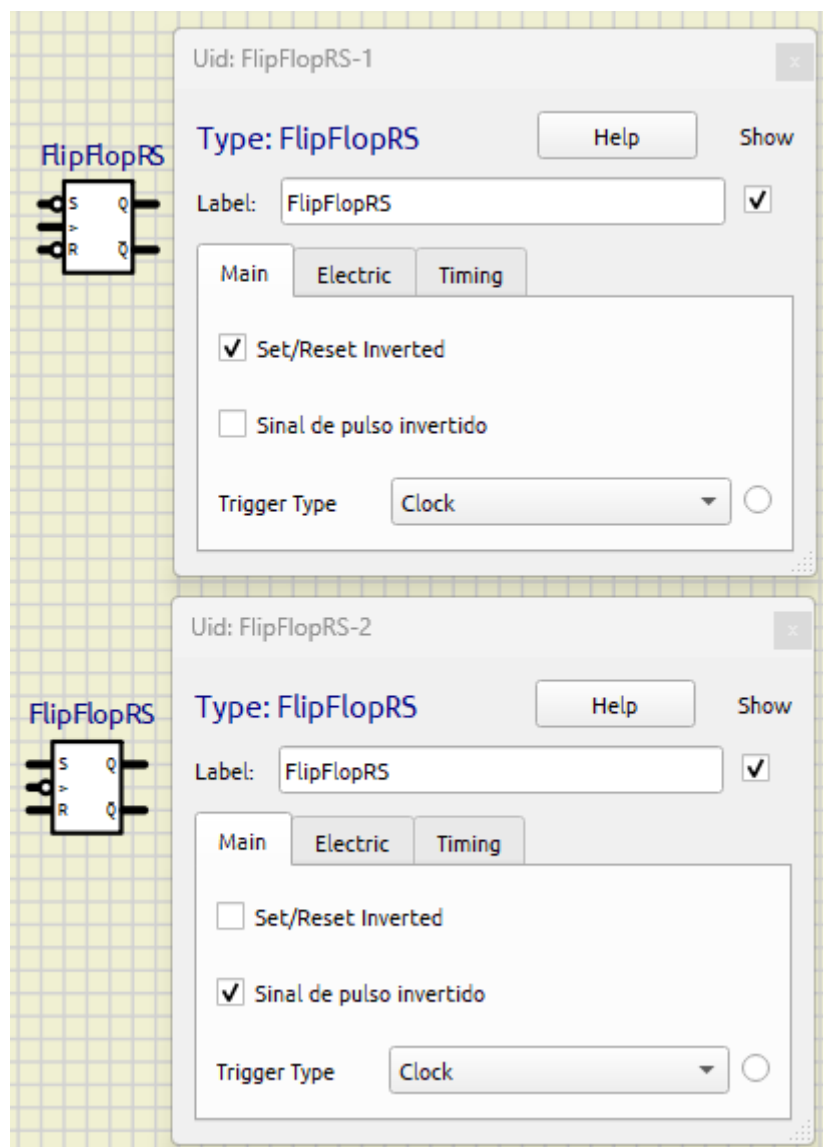
$S = 1$ e $R = 0 \rightarrow$ saída $Q = 1$ (set)

$S = 0$ e $R = 1 \rightarrow$ saída $Q = 0$ (reset)

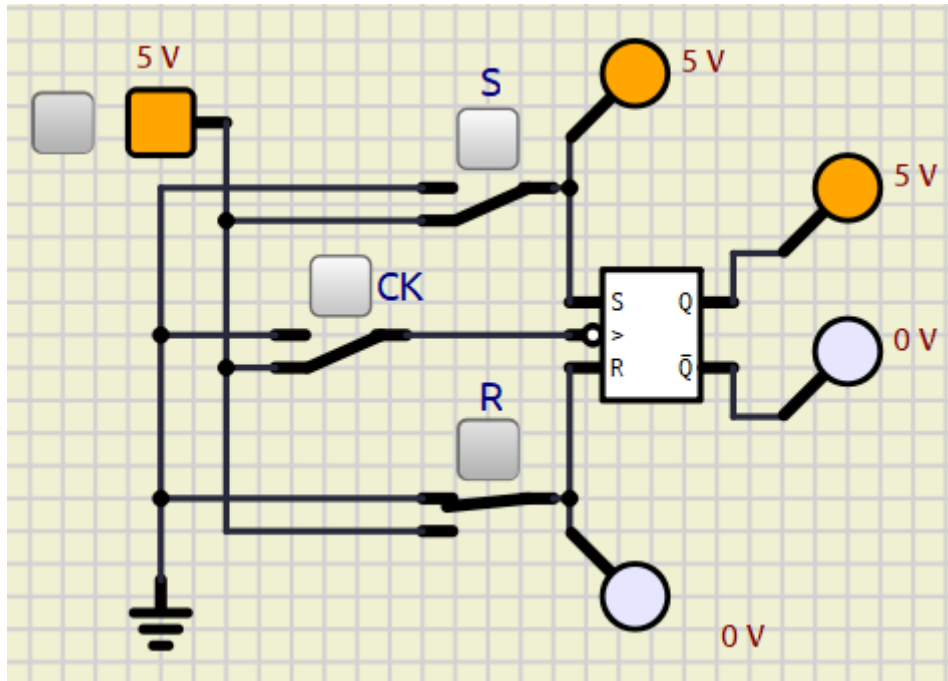
$S = 0$ e $R = 0 \rightarrow$ o estado anterior é mantido (hold)

$S = 1$ e $R = 1 \rightarrow$ instável (indefinido - saídas não se complementam)

As figuras abaixo mostram um FF tipo RS básico, sincronizados com pulsos de clock, onde o primeiro é sensível à subida (Rise) do pulso de clock enquanto que, o segundo é sensível à descida (Fall) do pulso de clock.



A figura abaixo mostra um FF RS operando no modo SET, pois $S = 1$ e $R = 0$, resultando em $Q = 1$ e $Q' = 0$.



Flip Flop JK:

O flip-flop JK é um dispositivo eletrônico digital e sequencial com a propriedade de armazenar um bit de informação (0 ou 1) e sincronizar suas mudanças de estado por meio de um sinal de clock.

É uma evolução do flip-flop SR (Set-Reset), corrigindo o maior defeito do FF RS quando as duas entradas (S e R) recebiam nível lógico 1 ao mesmo tempo, o circuito entrava em um estado inválido ou instável.

Portanto, não existe estado instável ou proibido no FF JK quando as entradas forem iguais a 1.

Veja abaixo os modos de operação do FF JK:

J = 0 e K = 0: Mantém o estado anterior (Hold / Memória)

J = 0 e K = 1: Coloca a saída Q em 0 (Reset)

J = 1 e K = 0: Coloca a saída Q em 1 (Set)

J = 1 e K = 1: Inverte o estado atual da saída Q (Toggle)

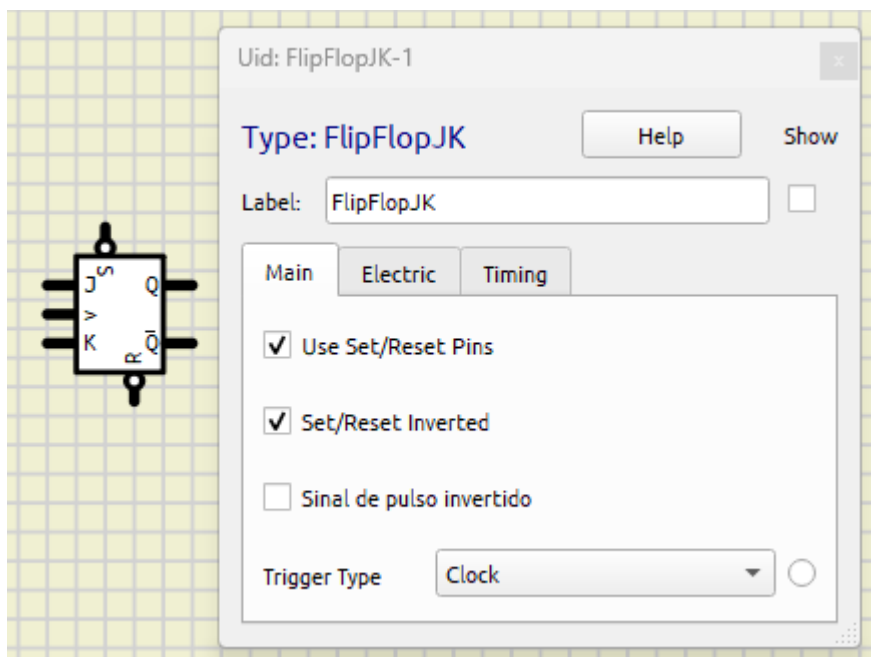
Principais Aplicações

Contadores digitais: O modo de inversão ou Toggle ($J=1$, $K=1$) faz dele a base ideal para contadores de pulsos.

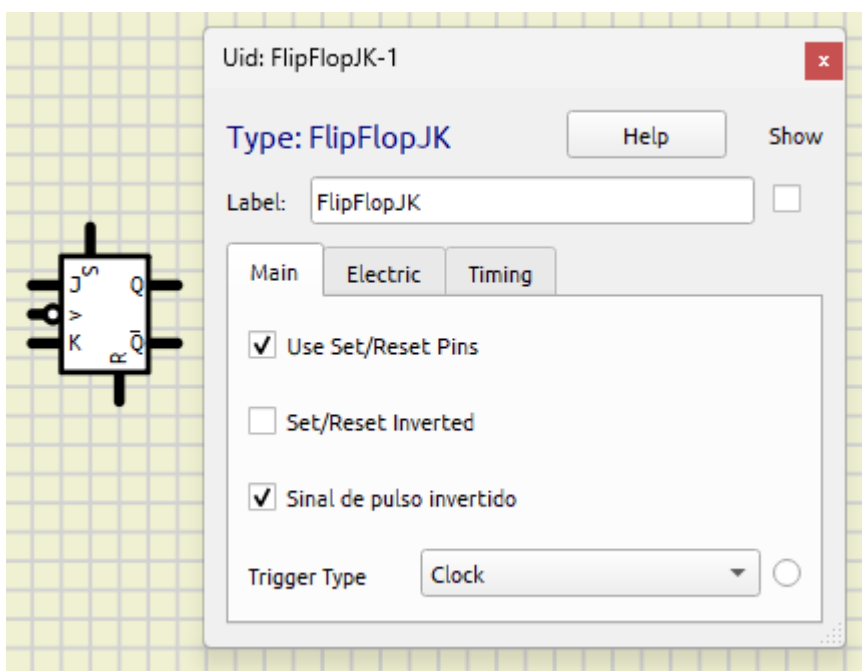
Registradores de deslocamento: Armazenamento temporário de sequências de bits.

Divisores de frequência: Reduzem a frequência de um sinal de clock pela metade a cada ciclo de inversão.

A figura abaixo mostra o padrão do FF JK ao ser inserido na área de trabalho, onde as entradas PR e CLR são ativas em 0 e a entrada de clock ativa em 1, ou seja, na subida do pulso de clock, denominada também transição L-H (low-high / baixa para alta).

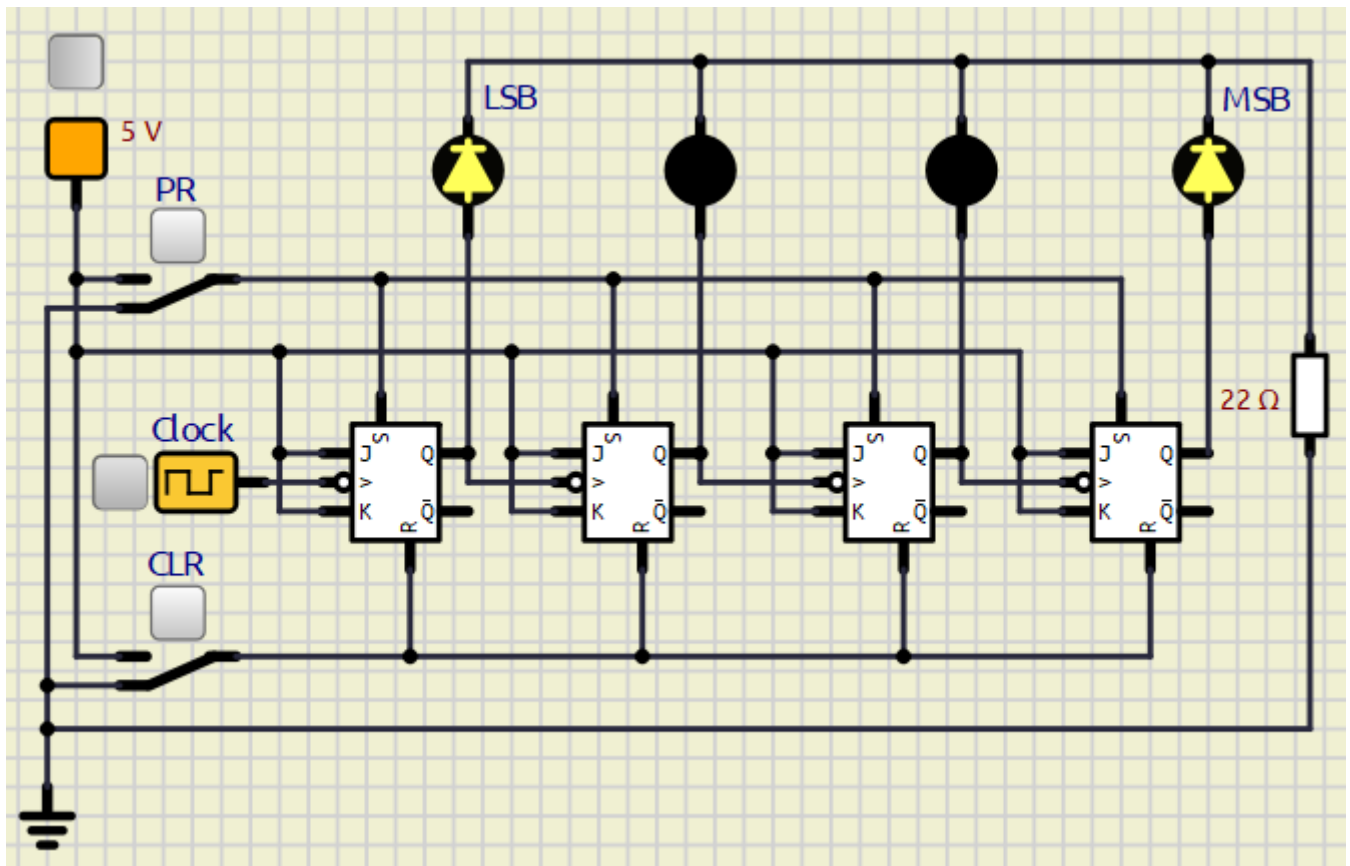


Na figura abaixo, as entradas PR e CLR são ativas em 1 e a entrada de clock ativa em 0, ou seja, descida do pulso de clock, denominada também transição H-L (high-low / alta para baixa).



A figura a seguir ilustra um contador de 4 bits, do tipo assíncrono progressivo, onde cada FF JK representa um bit.

Esse contador é denominado assíncrono pelo motivo do pulso de clock ser aplicado apenas no primeiro FF, que neste caso atua como LSB identificado como bit menos significativo.



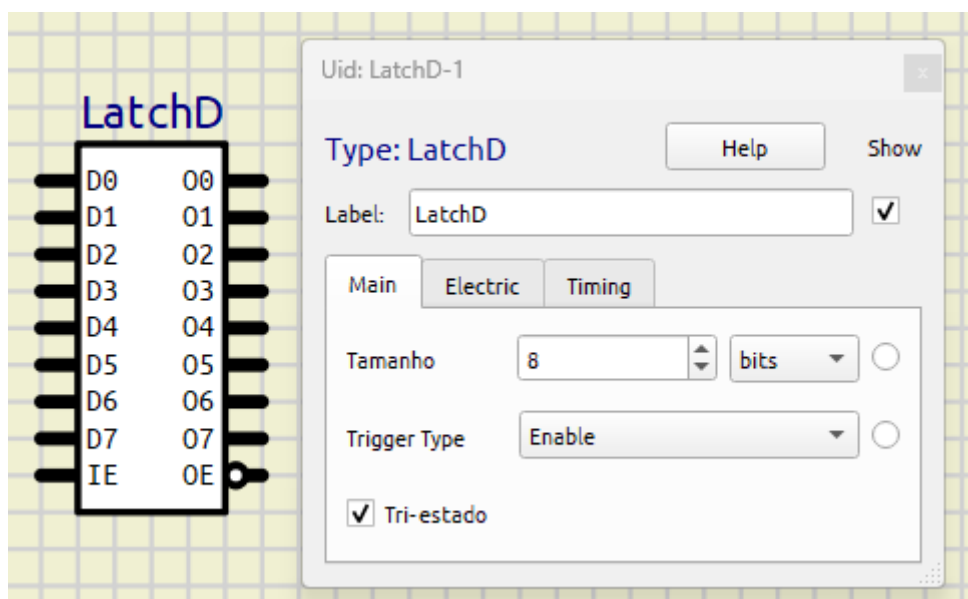
Todos os FFs estão operando no modo Toggle, visto que as entradas J e K estão em nível lógico 1.

Em BCD binário (BCD 8421), a contagem representa o número 9 decimal (1001).

Latch:

A função de um Latch é armazenar bit binário de forma temporária, operando como um elemento básico de memória digital.

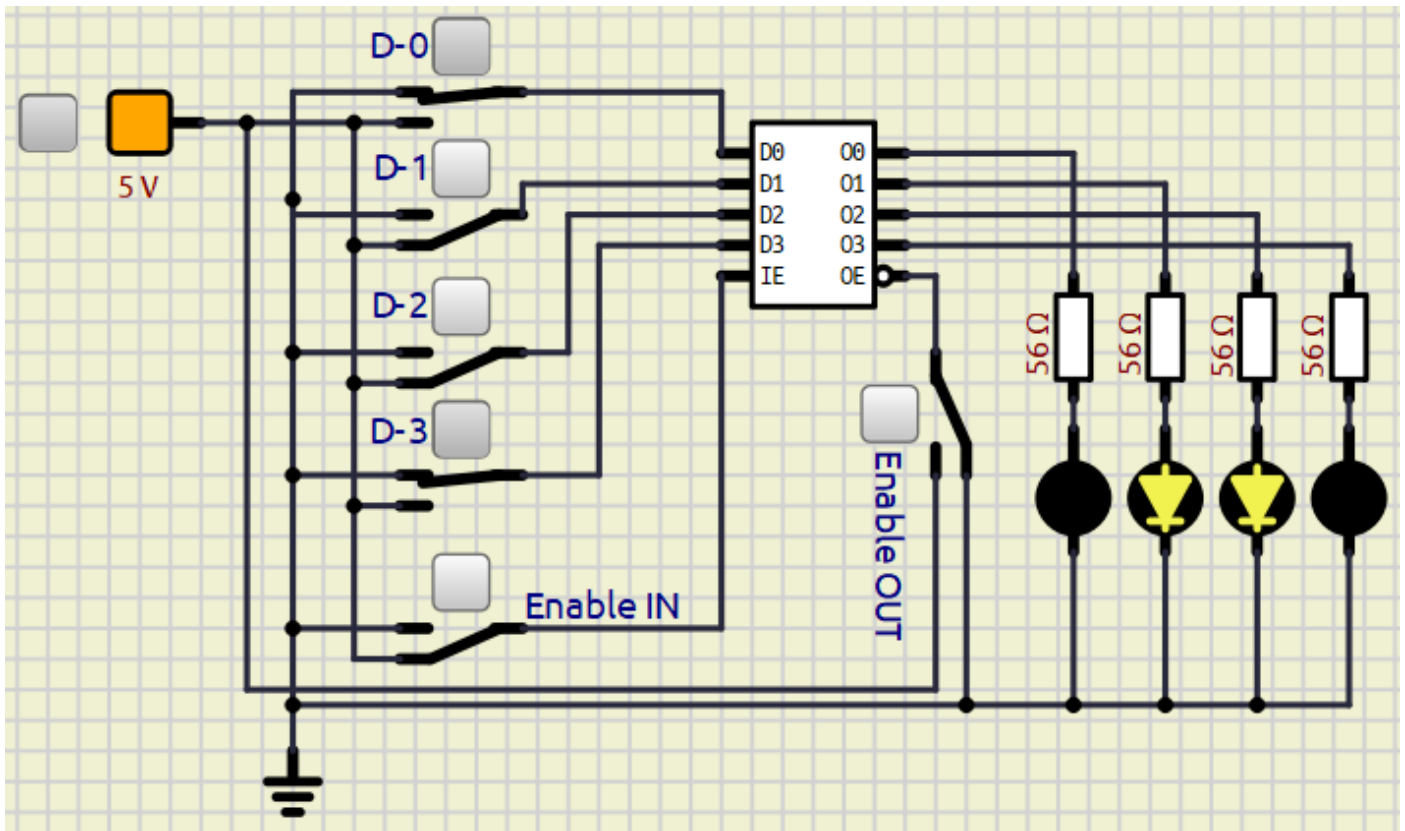
Por padrão, ao ser inserido no simulador o Latch vem configurado com 8 entradas e 8 saídas, podendo ser configuradas na opção "Tamanho".



Entrada IE: habilita em nível 0 ou 1 as entradas presentes (D0 a D7)

Saída OE: é um comando opcional que, libera as saídas para outros circuitos, ou seja, atua como um Tri-State (do inglês Tri-estado).

A figura a seguir ilustra a simulação de um Latch com 4 entradas:



Lógicos (Logic) - Conversores (Converters)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou botão direito do mouse – propriedades.

Multiplexador (Mux):

O Multiplexador (Mux), serve para combinar diversos canais de entrada para um canal de saída.

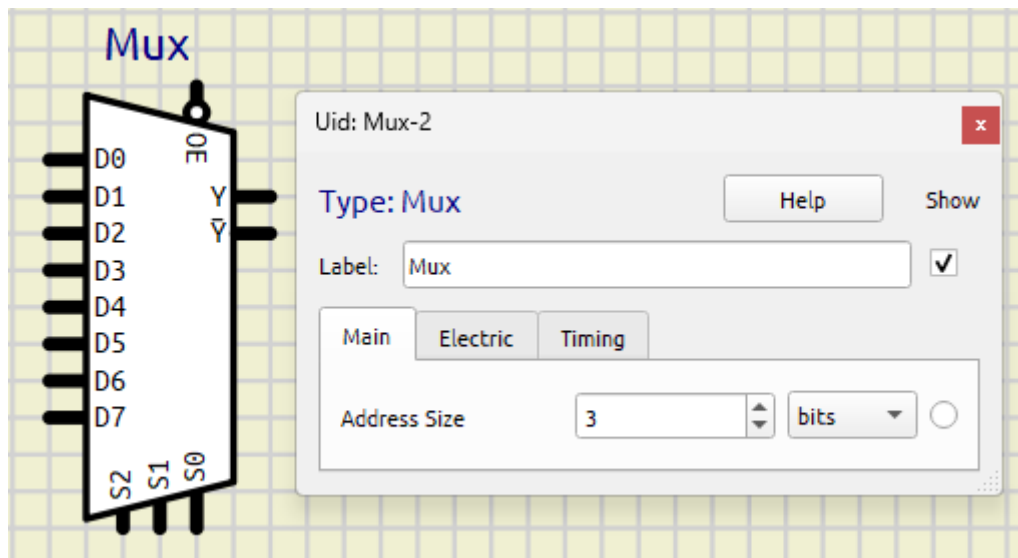
Essa seleção é feita através de um processo seletivo denominado endereçamento, funcionando assim, como uma chave seletora.

A quantidade de bits de endereçamento determina a quantidade de entradas para serem processadas.

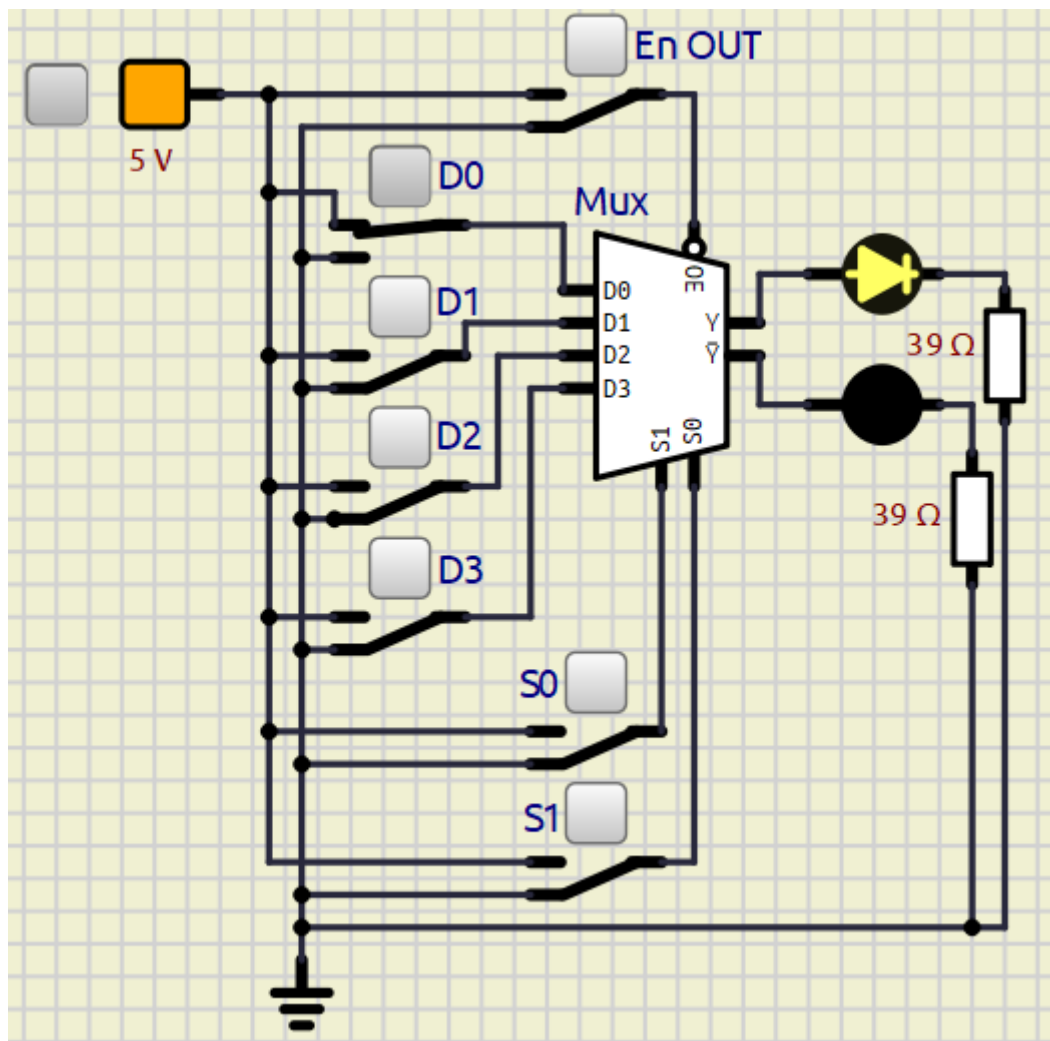
No caso, por padrão, o componente é inserido com endereço de 3 bits, o que significa que pode controlar 8 canais de entrada (2^3).

Um endereço com 2 bits controla 4 canais de entrada (2^2); um endereço com 4 bits controla 16 canais de entrada (2^4) e assim por diante.

A figura abaixo ilustra um Mux com configurações padrão do simulador, inserido na área de trabalho, com 3 bits de endereço, que pode ser modificado na opção "Address Size".

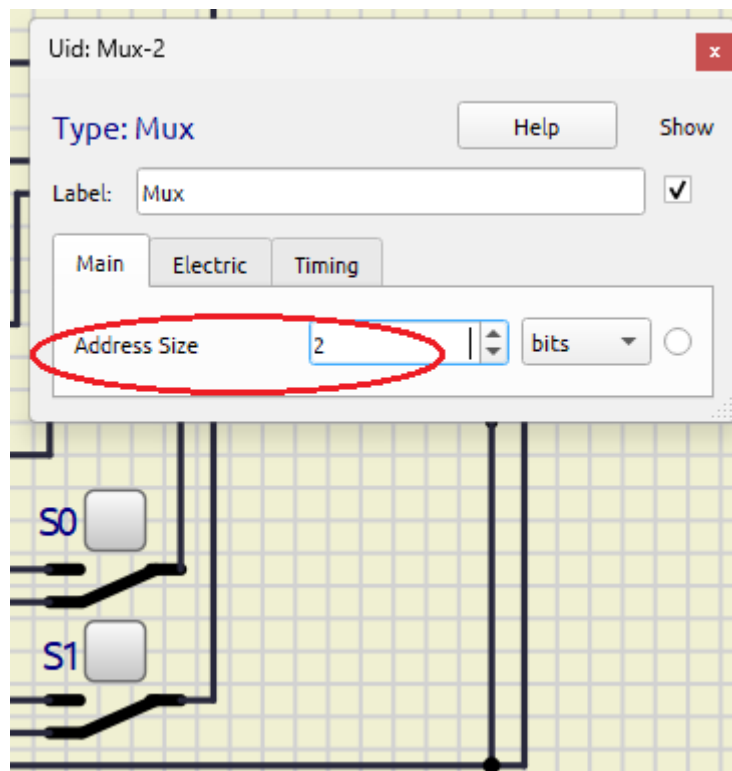


A figura abaixo mostra a simulação de um Multiplexador que possui 2 bits de endereçamento, permitindo o controle de 4 entradas (2^2):



A entrada D0 está sendo encaminhada ao canal de saída Y, pelo endereçamento igual a 00 ($S0 = 0$ e $S1 = 0$).

Veja que a opção "Address Size" está ajustada para 2 bits

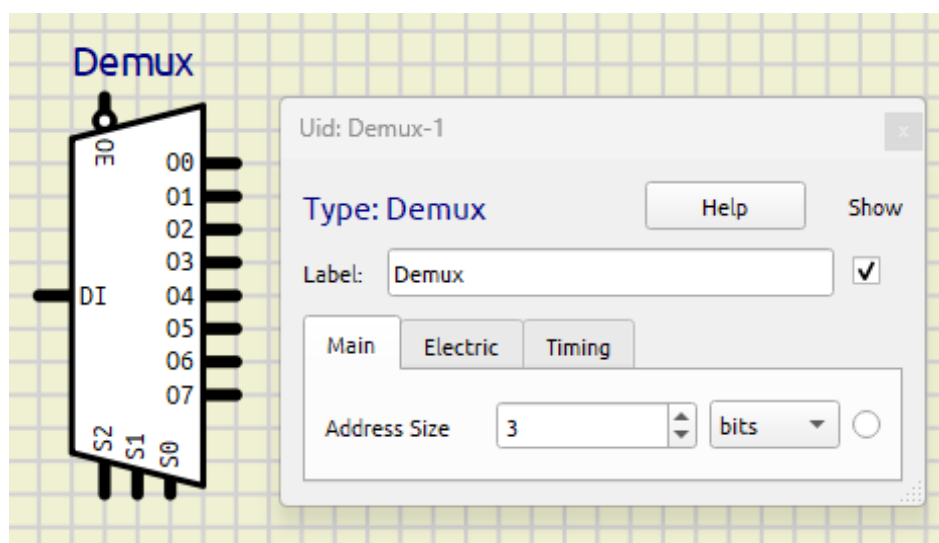


Demultiplexador (Demux):

O demultiplexador (DEMUX) ao contrário de um multiplexador (MUX), recebe um único sinal de entrada e o direciona seletivamente para uma entre várias linhas de saída, através de um endereçamento.

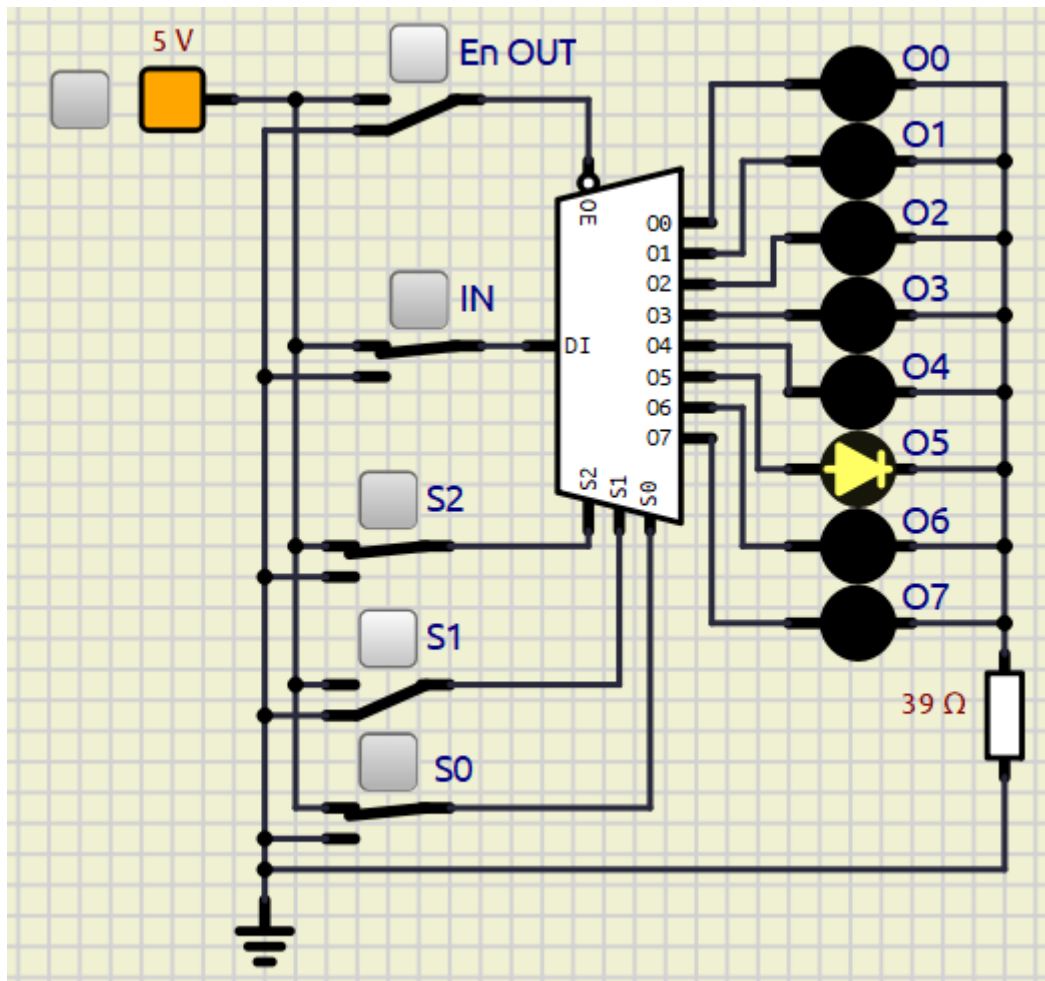
Da mesma forma que no MUX, a quantidade de bits na entrada de endereçamento determina a quantidade de canais de saída.

Por padrão, o simulador insere na área de trabalho o Demux com 3 bits de endereço, que pode ser modificado na opção "Address Size".

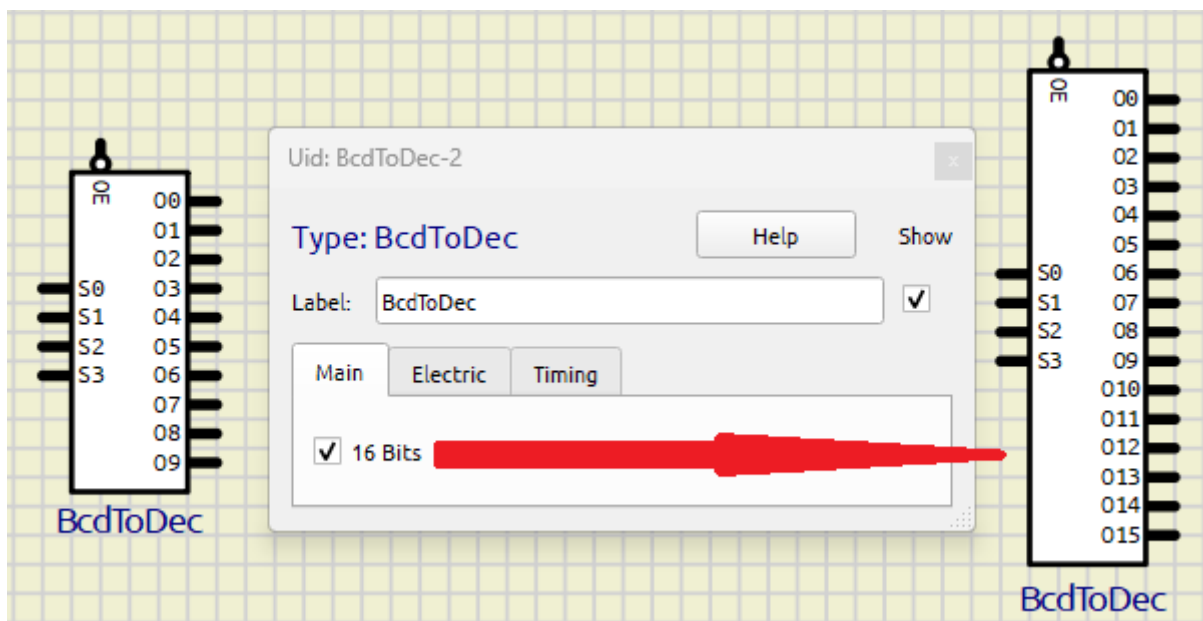


A figura abaixo ilustra a simulação de um Demux de 8 canais de saída, com endereçamento de 3 bits.

Observe que para o endereçamento 101 (que corresponde a 5 decimal) o sinal de entrada é dirigido à saída O5.



Decodificador (4 para 10/16) BcdToDec:



Trata-se de um decodificador BCD 8421 para 10 ou 16 bits (linhas) na saída, ou ainda, pode ser definido como um decodificador BCD 8421 para decimal ou hexadecimal. Daí a especificação de 4 para 10/16.

As informações inseridas em S0, S1, S2 e S3, correspondem ao BCD 8421 e são enviadas às saídas 09 ou 015, dependendo se na opção "Main" a caixa de seleção (checkbox) "16 Bits" estiver ou não selecionada.

Por padrão do simulador, a inserção está com a opção "16 Bits" desmarcada e, portanto, a conversão será: 4 para 10 (4 para 10 linhas).

Na opção 4 para 10, o máximo direcionamento será obtido na inserção do código BCD 1001, que representa o decimal 9.

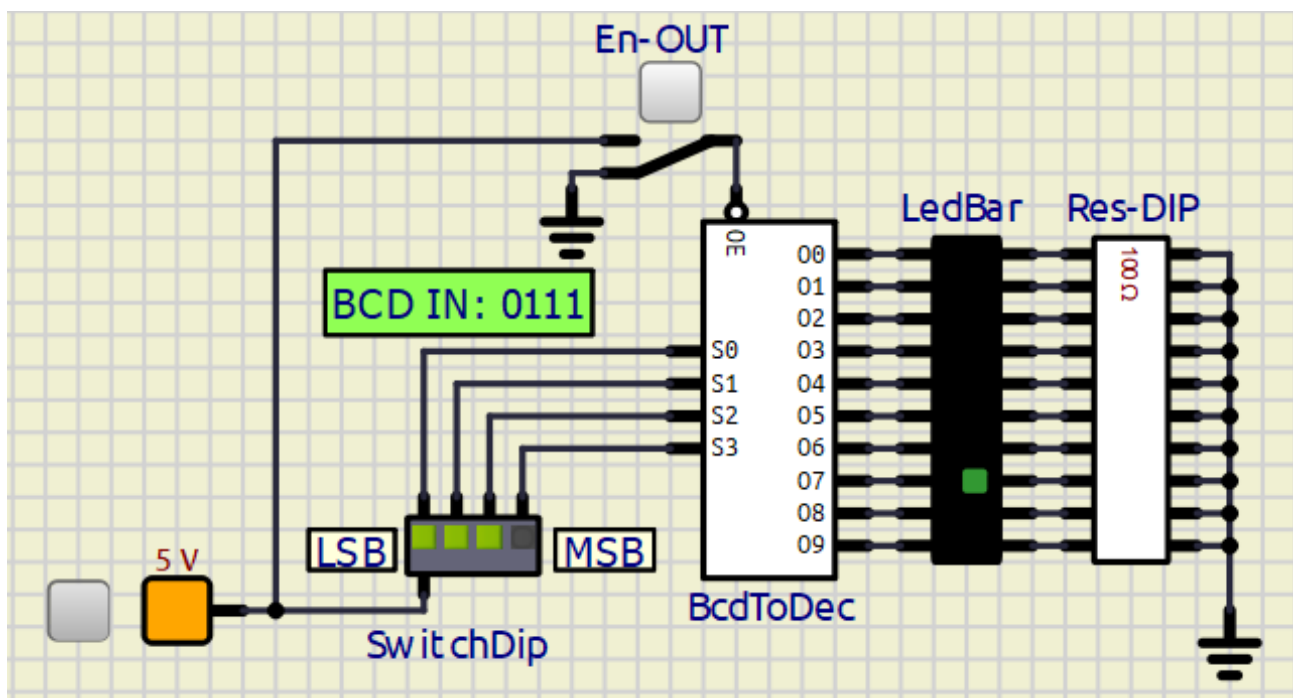
Acima disso, ou seja, a partir de 1010 não haverá nenhuma atividade nas saídas. Tal não ocorre se a opção "16 Bits" estiver ativada.

Resumindo:

4 para 10: 0000 a 1001 (0 a 9)

4 para 16: 0000 a 1111 (0 a 15)

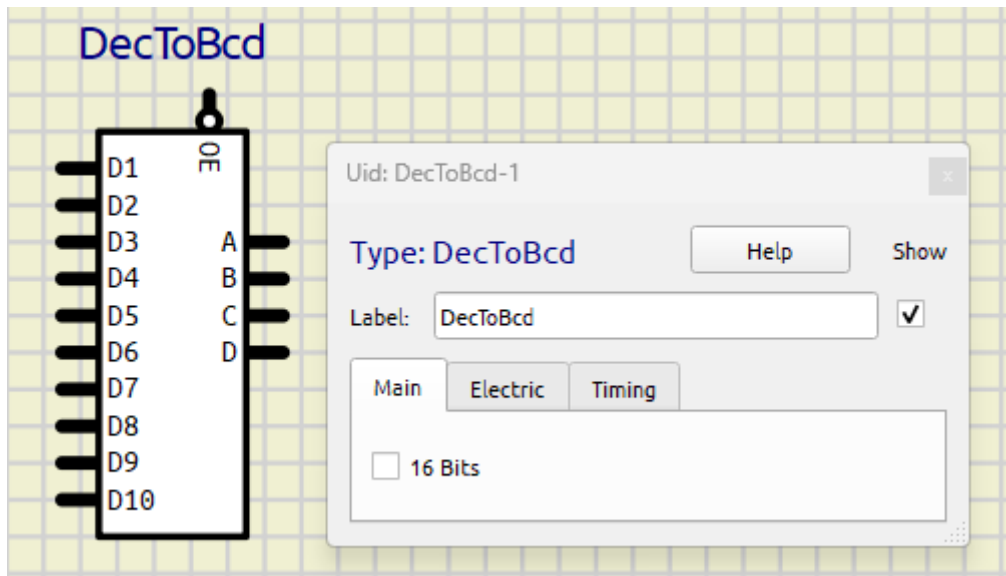
A figura abaixo ilustra a simulação de um decodificador BCD 4 para 10, onde nas entradas S0 a S3 é introduzida a informação binária 0111 (BCD), que aparece decodificada na saída 07 (equivalente decimal).



Codificador (10/16 para 4) DecToBcd:

O codificador opera de forma contrária, ou seja, a partir de uma informação correspondente em decimal, converte a saída em BCD, logo, podemos dizer que se trata de um codificador de decimal para BCD.

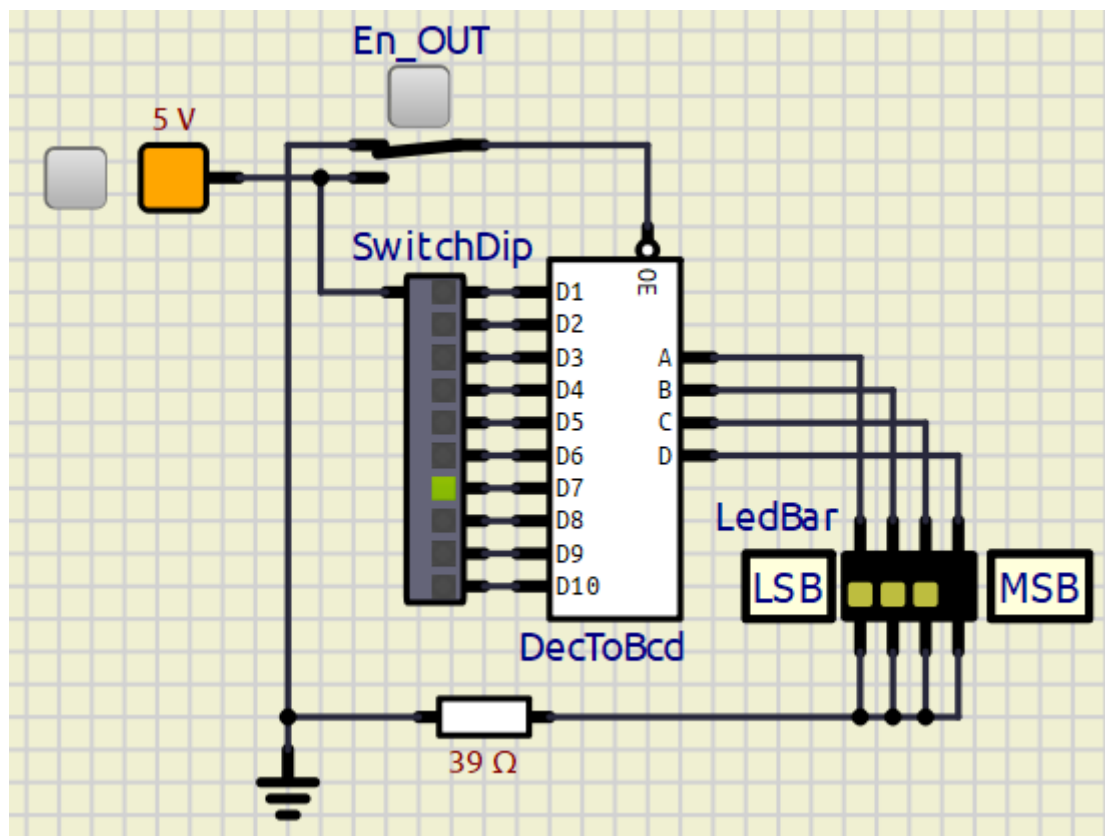
Da mesma forma que no decodificador visto anteriormente, na opção "Main" a caixa de seleção (checkbox) permite selecionar também a codificação de 16 bits para 4 bits (ou 16 linhas para 4 linhas), uma vez que por padrão, a inserção do componente é 10 para 4.



A figura abaixo mostra um codificador 10 to 4, ou seja, decimal para BCD, onde as entradas D1 a D10 correspondem a equivalência BCD 0000 a 1001.

A entrada D10, devido a uma limitação teórica do BCD (0000 a 1001) não deve funcionar, pois para mostrar em BCD legítimo seriam necessários 2 bits separados:

0001 0000 (BCD) → decimal 10

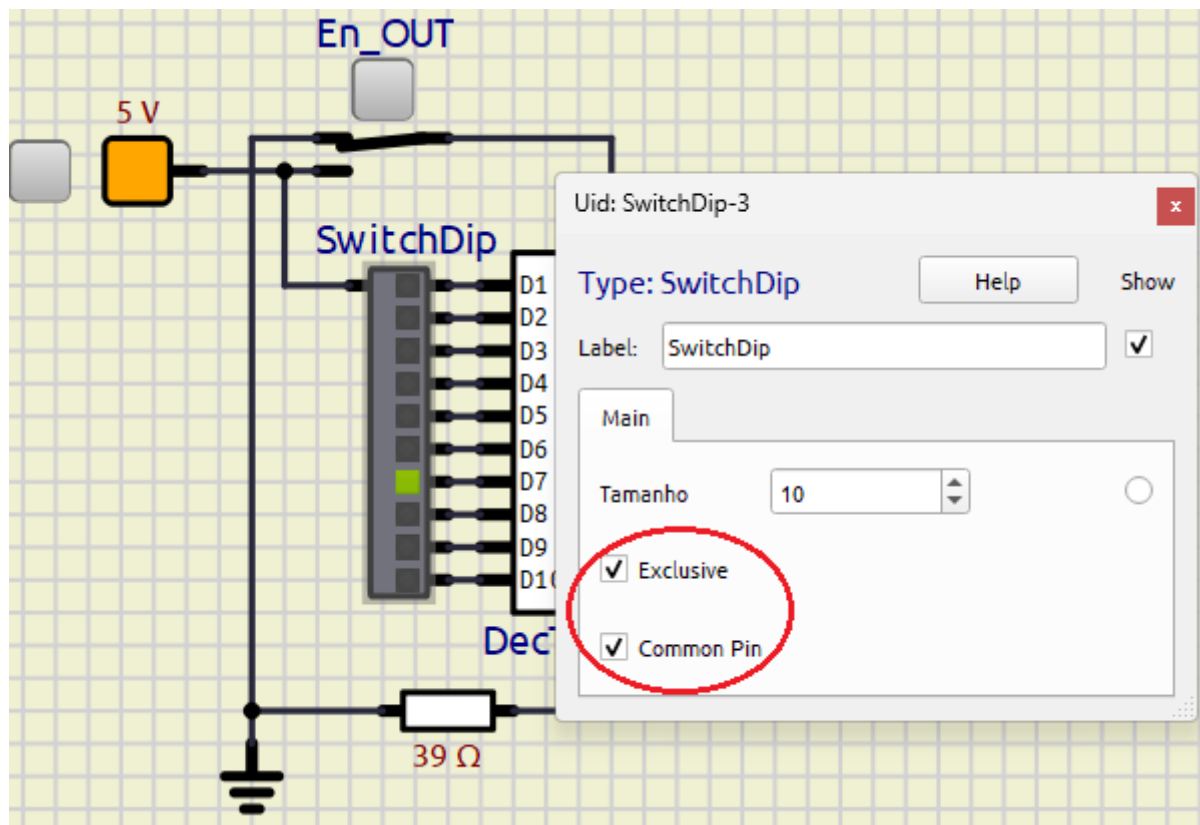


Observe que a entrada selecionada D7, aparece codificada BCD na saída: 0111, que equivale ao decimal 7.

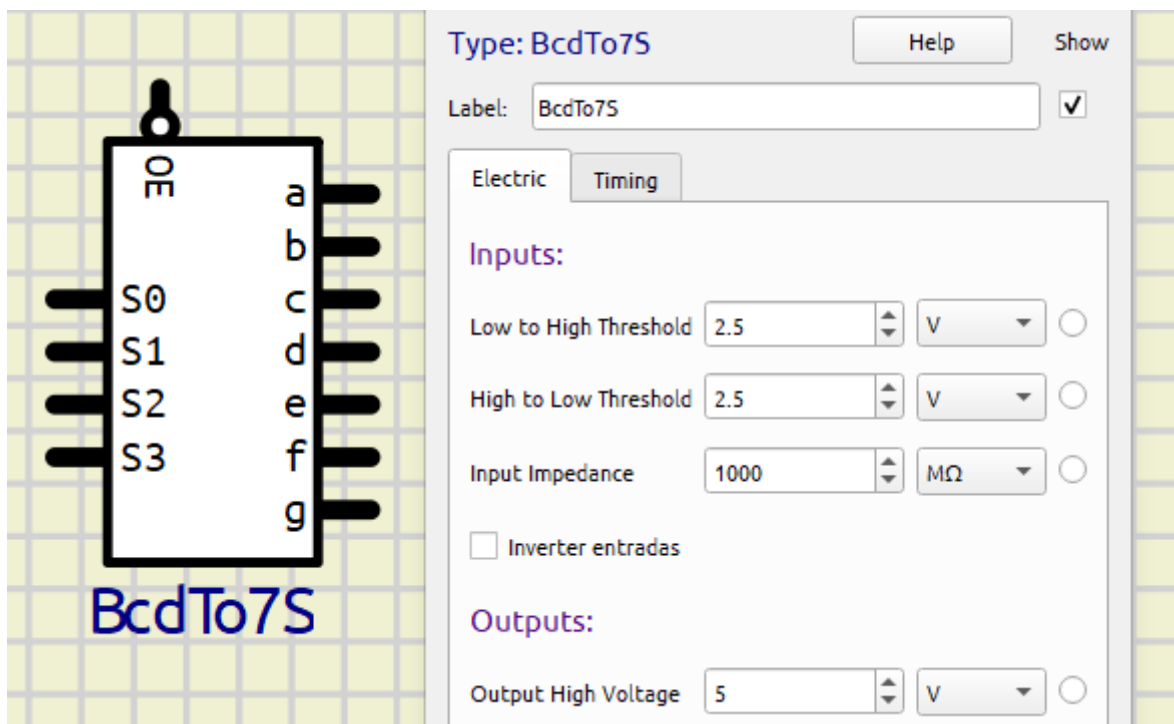
Para uma melhor performance durante a simulação, é recomendável selecionar as duas caixas de seleção na opção "Main"

Exclusive: ativa a última entrada selecionada, desativando a existente ou selecionada anteriormente

Common Pin: interliga internamente todos os de entrada

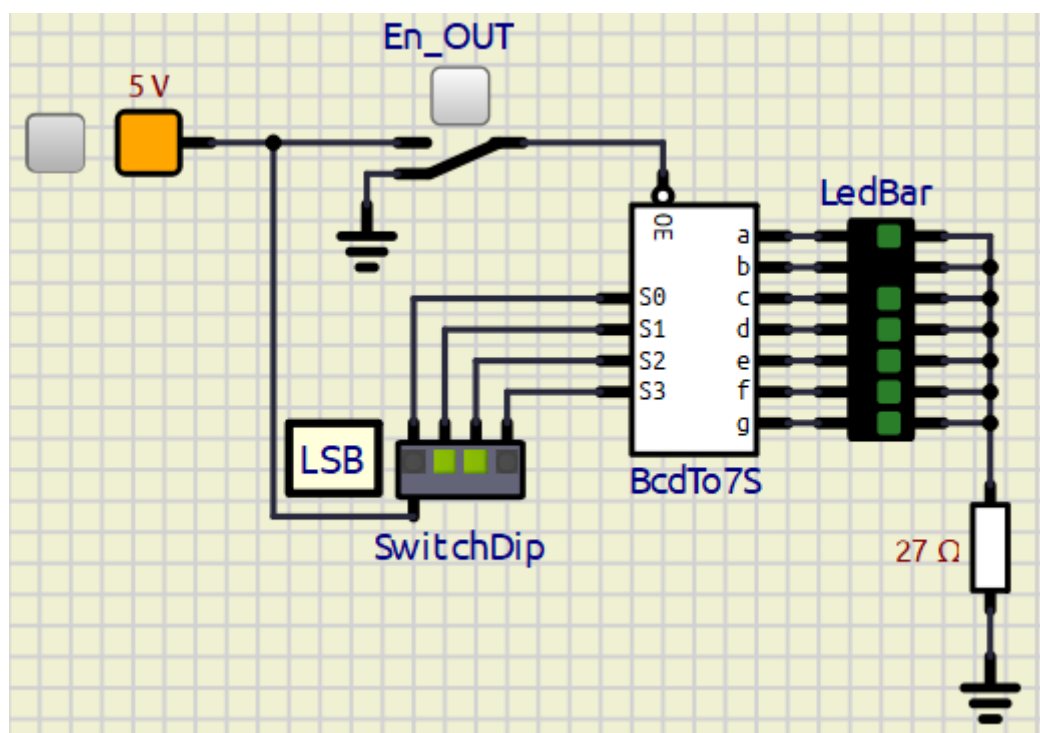


BCD para 7 segmentos BcdTo7S:



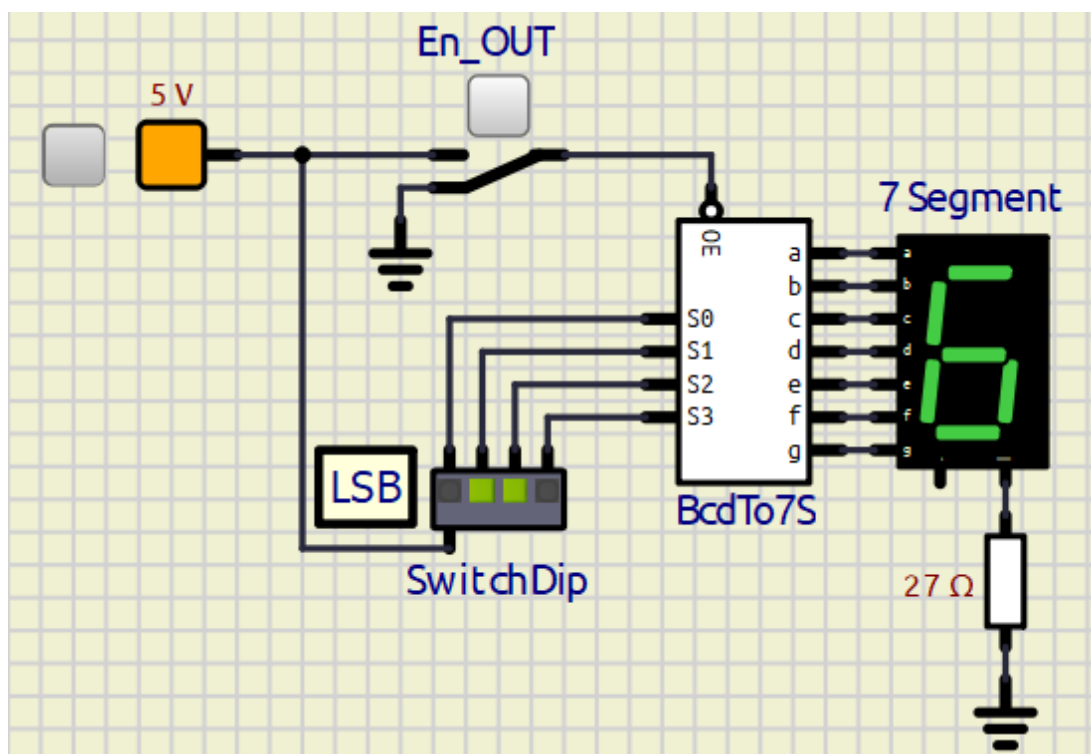
Esse decodificador conforme o nome sugere, converte uma entrada BCD 8421 para um display de 7 segmentos.

A figura abaixo ilustra uma montagem para simulação, onde é introduzida a expressão binária BCD 0110 nas entradas S0 a S3, ativando nas saídas as letras que correspondem ao número decimal 6.



Na figura abaixo, temos as mesmas condições anteriores, porém o componente LedBar, foi substituído por um display de 7 segmentos, de catodo comum.

Na opção "Electric" do display, ao desmarcar na caixa de seleção a opção "Common Cathode", o display assume a polarização de anodo comum.



O sinal negativo (-) onde está conectado o resistor de 27Ω, indica tratar-se de display de catodo comum, caso contrário o sinal seria positivo (+).

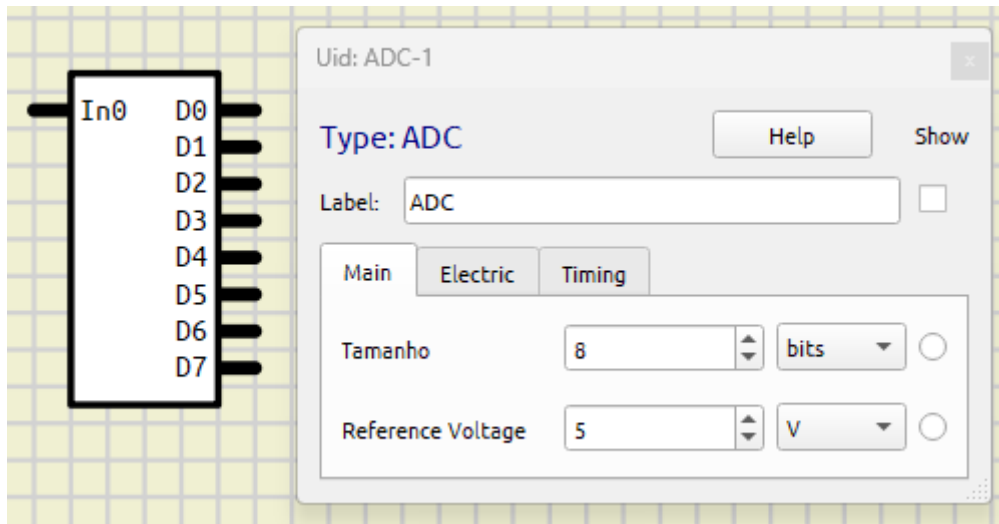
Lógicos (Logic) - Outros lógicos (Other Logic)

Para configurar as propriedades de cada componente, dar um duplo clique no ícone correspondente ou **botão direito do mouse – propriedades.**

Conversor analógico-digital (ADC):

A figura abaixo ilustra o componente inserido na área de trabalho.

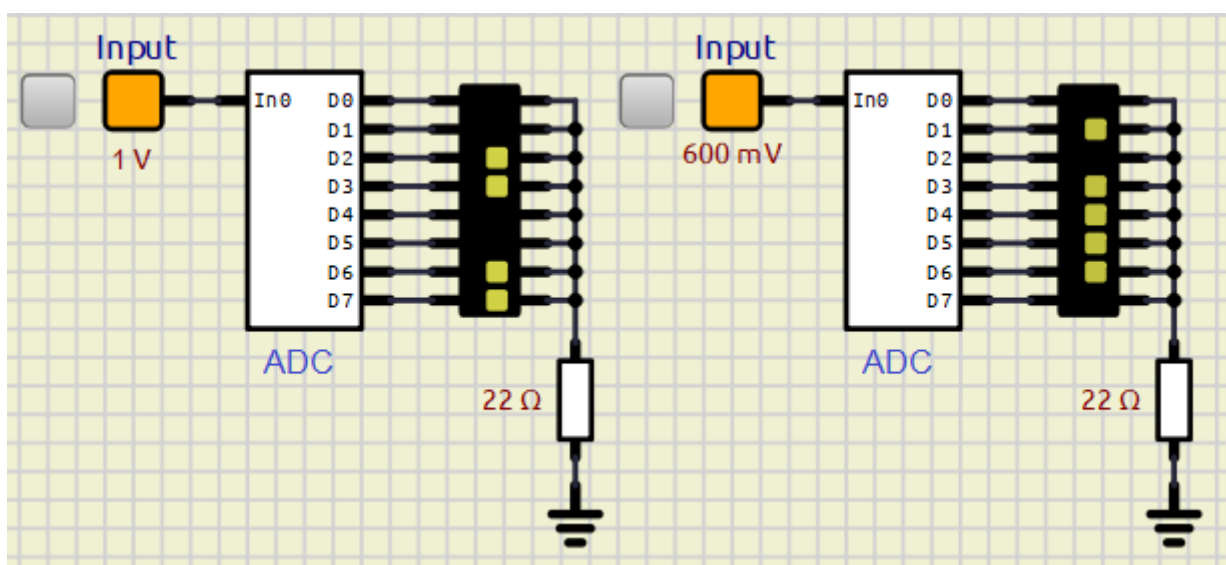
Por padrão as saídas D0 a D7 não são invertidas, daí caso necessário, na opção "Electric" selecionar a caixa de seleção "Invert Outputs".



Na entrada In0 um sinal analógico aplicado na entrada e nas saídas (D0 a D7) converte o mesmo em sinal digital.

Esse sinal analógico pode ter origem em um sensor de temperatura, de luz, de posição, etc. de tal forma a ser convertido em digital (0 ou 1) para que possa ser processado, por exemplo, por um microcontrolador.

A figura abaixo ilustra a simulação de um Conversor analógico-digital de 8 bits, com aplicação de sinais analógico na entrada de **1V e 600mV**.



Procedendo a leitura e considerando a saída "D0" LSB, temos:

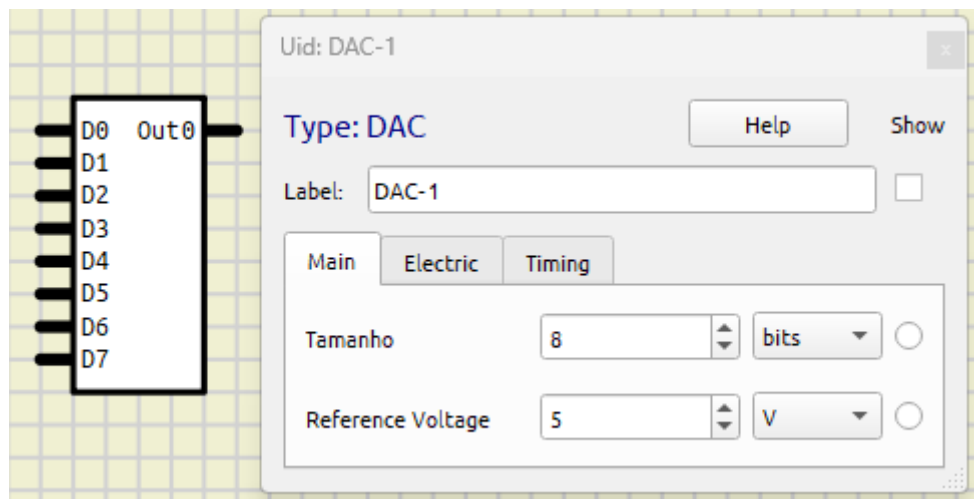
Sinal analógico: 1V → Saída digital: 11001100

Sinal analógico: 600mV → Saída digital: 01111010

Conversor digital-analógico (DAC):

A figura abaixo ilustra o componente inserido na área de trabalho.

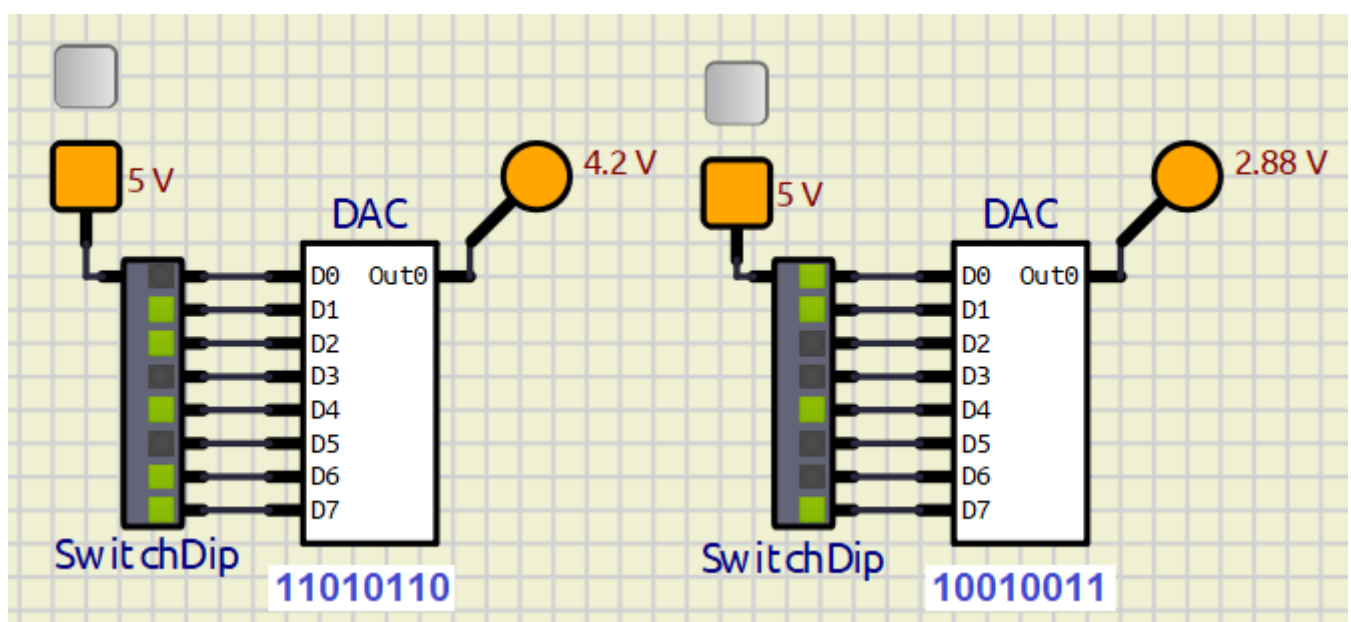
Por padrão as entradas D0 a D7 não são invertidas, daí caso necessário, na opção "Electric" selecionar a caixa de seleção "Inverter entradas".



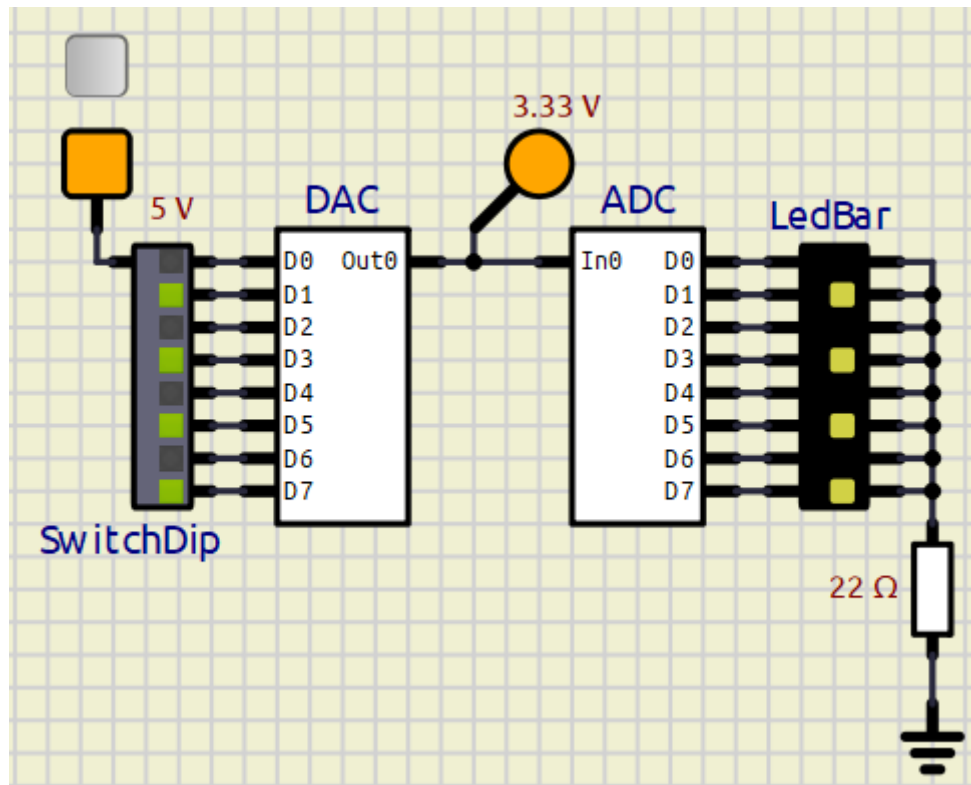
O sinal digital aplicado na entrada é convertido em analógico na saída "Out 0".

A partir de uma combinação de sinais digitais (0 ou 1) em um teclado, permite gerar sinais analógicos para acionamento remoto de dispositivos, como por exemplo, uma trava eletrônica.

A figura abaixo ilustra a simulação de um Conversor digital-analógico de 8 bits, com aplicação de sinais digitais nas entradas: **11010110 e 10010011**, considerando a entrada "D0" LSB.



A figura abaixo ilustra a simulação entre a ligação de um Conversor digital-analógico e um Conversor analógico-digital:

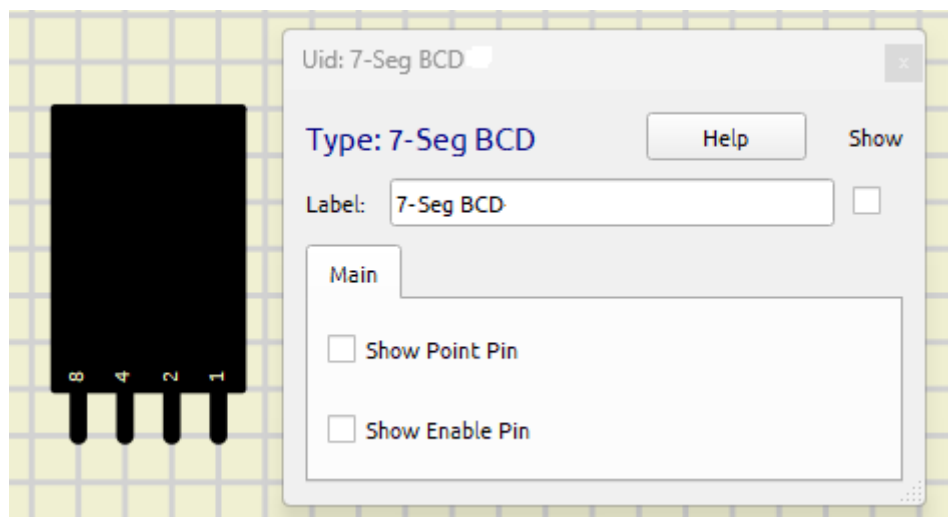


Entrada digital: 10101010

Saída digital: 10101010

7 segmentos BCD (7-Seg BCD):

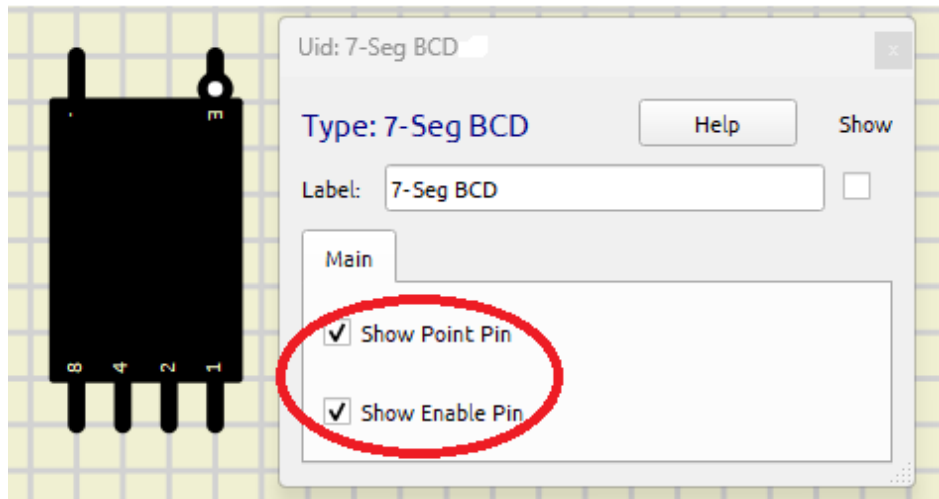
Esse display já possui um decodificador BCD para 7 segmentos integrado internamente, daí então possuir 4 terminais (BCD8421).



A figura acima mostra o display com identificação dos pinos em 8421, que é exatamente a entrada de um decodificador BCD 8421, porém, neste display esse decodificador já está internamente incorporado.

Por padrão do simulador esse display é inserido sem as opções de mostrar o ponto e o pino de habilitar os segmentos.

Para isso, basta marcar as caixas de seleção "Show Point Pin" e "Show Enable Pin".



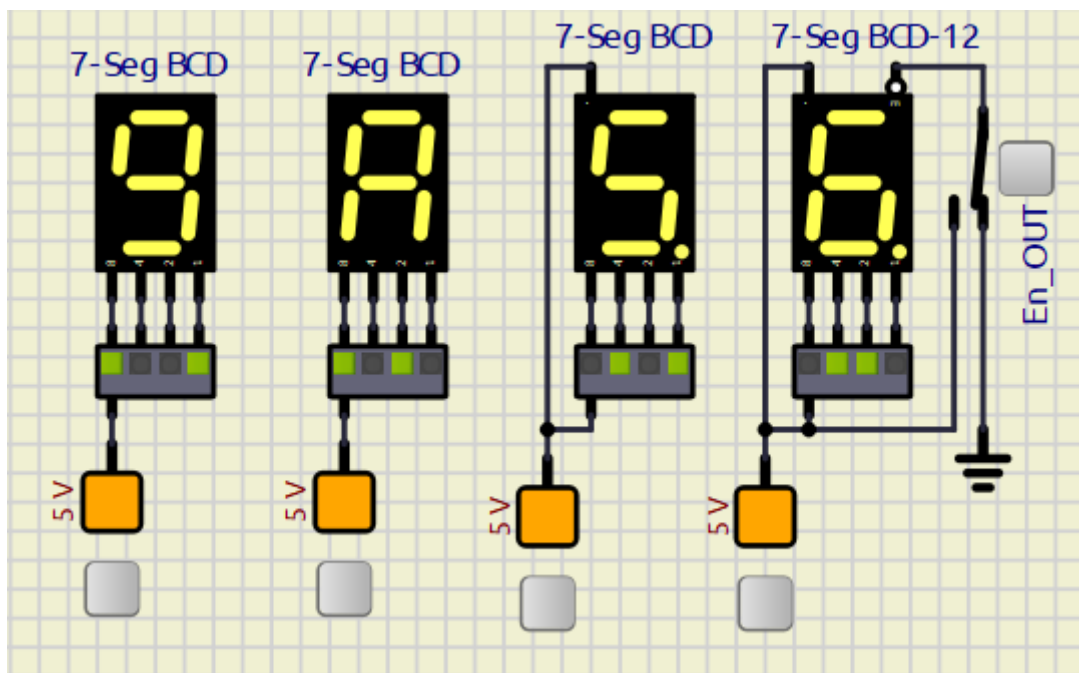
Veja na figura abaixo a simulação com as seguintes condições:

Entrada BCD 8421: 1001 → saída decimal "9"

Entrada BCD 8421: 1010 → saída hexadecimal "A"

Entrada BCD 8421: 0101 → saída decimal 5.

Entrada BCD 8421: 0110 → saída decimal 6.



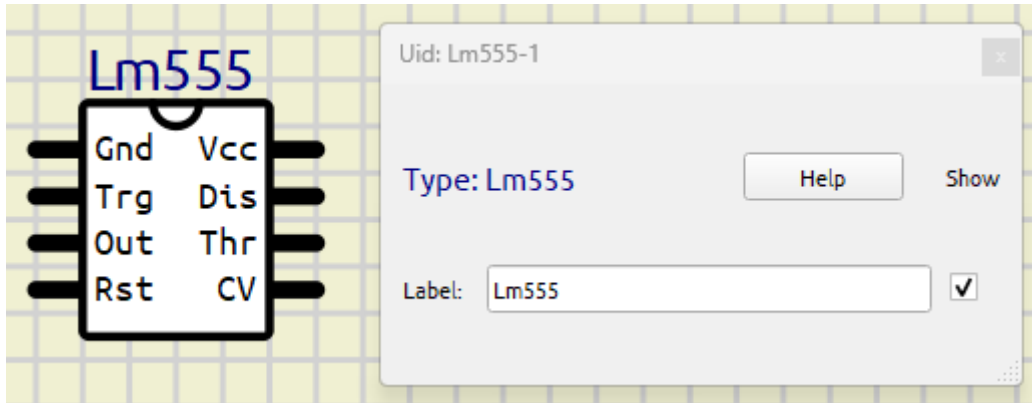
OBS:

a) no display cuja entrada BCD é 0101 (decimal 5.), aparece o ponto (DOT) por estar esse terminal conectado à fonte de 5V.

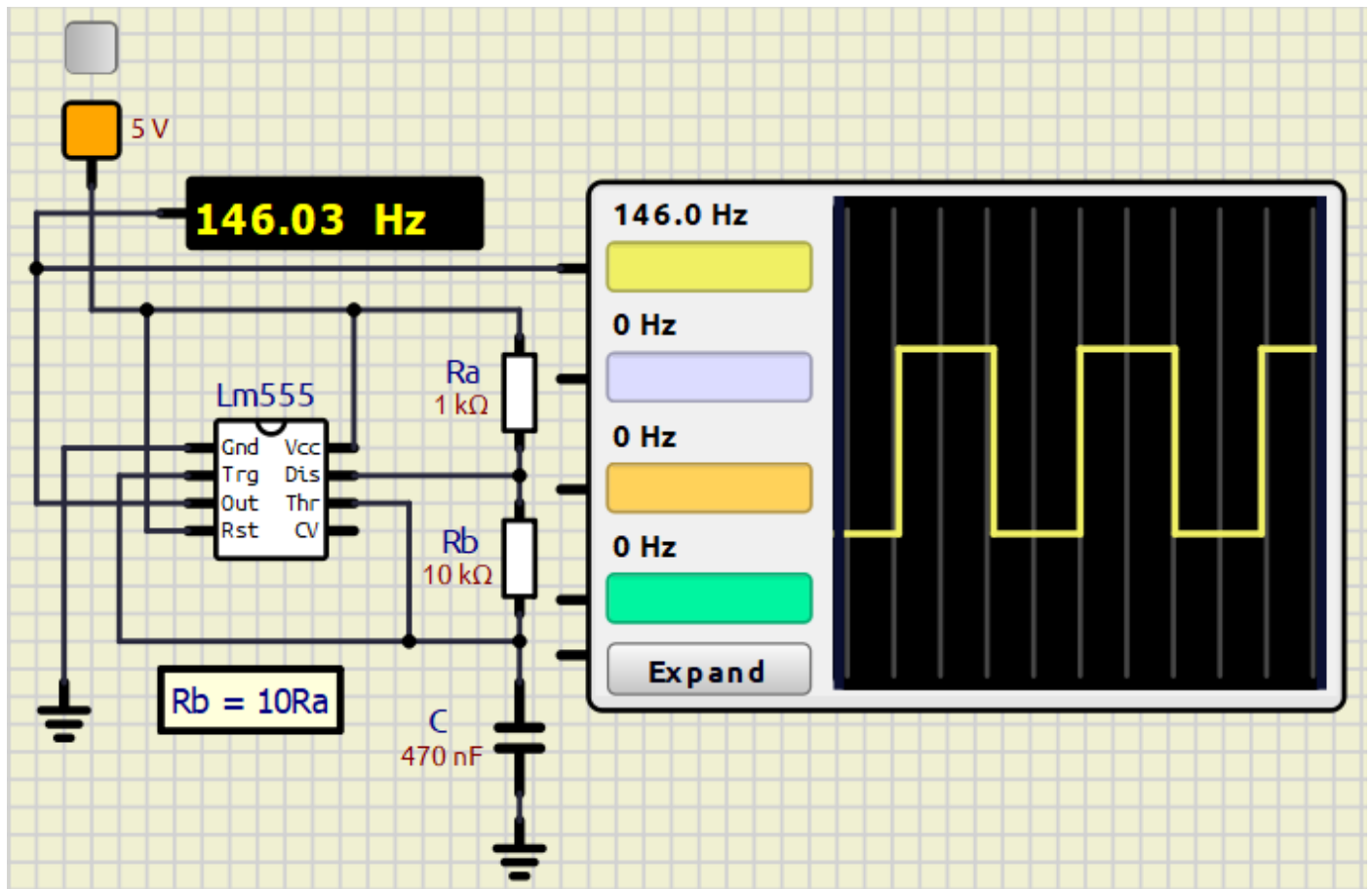
b) no display cuja entrada BCD é 0110 (decimal 6.), além do terminal do ponto, o terminal para habilitar/permitir a visualização no display também está habilitado, uma vez que, por ser ativo em 0, está devidamente aterrado.

LM555:

O circuito integrado LM555 é basicamente um temporizador (timer), que pode operar na simulação de circuitos digitais como um oscilador gerando onda quadrada (modo astável), que pode ser utilizada como pulsos ou sinais de clock para sincronização de circuitos digitais, do tipo contadores e registradores de deslocamento.



A figura abaixo mostra a construção e simulação de um gerador de clock:



A frequência de oscilação pode ser alterada, modificando o valor do capacitor de 470nF. O valor da capacitância é inversamente proporcional à frequência, ou seja, diminuindo o valor de sua capacitância, a frequência aumenta.

$$f = 1,44 / [(Ra + 2Rb) \times C]$$
$$C = 1,44 / [(Ra + 2Rb) \times f]$$

ÍNDICE

Parte II - Eletrônica Digital

DESCRIÇÃO	PÁGINA
MEDIDORES (Meters)	
Analizador Lógico (Logic Analyzer / LAnalyzer)	01
FONTES (Sources)	
Gerador de pulso (Clock)	03
INTERRUPTORES (Switches)	
Teclado (KeyPad)	04
SAÍDAS (Outputs)	
7 segmentos (Seven Segment)	05
Matriz de LED (LedMatrix)	07
ATIVO (Active)	
Multiplexador Analógico (MuxAnalog)	10
LÓGICOS (Logic) - Portas (Gates)	
Buffer	12
Porta AND (AND Gate)	14
Porta OR (OR Gate)	16
Porta XOR (XOR Gate)	17
LÓGICOS (Logic) - Aritméticos (Arithmetic)	
Binary Counter	20
Somador Completo (Full Adder)	21
Shift Register	22
Função (Function)	24
LÓGICOS (Logic) - Memórias (Memory)	
Flip Flop D	26
Flip Flop T	27
Flip Flop RS	29
Flip Flop JK	30
Latch	32
LÓGICOS (Logic) - Conversores (Converters)	
Multiplexador (Mux)	33
Demultiplexador (Demux)	35
Decodificador (4 para 10/16) - BcdToDec	36
Codificador (10/16 para 4) - DecToBcd	37
BCD para 7 segmentos (BcdTo7S)	39
LÓGICOS (Logic) - Outros lógicos (Other logic)	
Conversor analógico-digital (ADC)	41
Conversor digita-analógico (DAC)	42
7 segmentos BCD (7-Seg BCD)	43
LM555	45