

1. Portas lógicas — símbolo, expressão e tabela-verdade

AND

$S = A \cdot B$

XOR (ou-exclusivo)

$S = A \oplus B$

OR

$S = A + B$

XNOR (coincidência)

$S = \overline{A \oplus B}$

NAND

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

NOT (inversor)

$S = \overline{A}$

NOR

$S = \overline{A + B}$

BUFFER

$S = A$

A	B	AND	OR	NAND	NOR	XOR	XNOR
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	0

a bolha ◦ na saída inverte: NAND é AND+bolha, NOR é OR+bolha, XNOR é XOR+bolha. Bolha na entrada inverte aquele sinal antes de ele entrar na porta.

Complemento: $\overline{A}$ no desenho, A' quando o traço não for viável (e-mail, código). **NAND e NOR são universais**: qualquer função sai só com uma delas.

2. Convenções de leitura

Sentido do fluxo. Entradas à esquerda, saídas à direita; *clock* e controle por baixo ou por cima. Lê-se como texto.

Ativo em 0. A barra sobre o nome ($\overline{\text{CLR}}$) e a bolha no símbolo dizem a mesma coisa: o sinal age quando vale **0**. Bolha na entrada de uma porta equivale a um inversor ali.

Numeração de bits. O bit **0** é sempre o menos significativo. $A[7:0]$ é um barramento de oito bits; A_3 é o bit 3.

Nomes de sinal. Em maiúsculas. Complemento com barra no desenho ($\overline{Q}$) e apóstrofo no texto corrido (Q') — nunca os dois juntos.

5. Seleção e decodificação

MUX 4×1

a seleção escolhe uma entrada e a copia na saída

DEMUX 1×4

leva a entrada para uma das saídas; as outras ficam em 0

Decodificador

n entradas $\rightarrow 2^n$ saídas; só uma vai a 1

Codificador

caminho inverso: a entrada ativa vira seu número em binário

O trapézio é a forma canônica de MUX e DEMUX: a base maior fica do lado com mais linhas e a seleção entra pela base inclinada, nunca junto com os dados.

3. Fios, junções e barramentos

ponto cheio = conectados

sem ponto = apenas se cruzam

barramento: traço grosso e largura em bits

splitter: separa ou junta os bits de um barramento

túnel: mesmo rótulo = mesmo nó, sem fio desenhado

tri-state: $EN=0$ desliga a saída (alta impedância, Z)

pull-up: define 1 quando nada aciona a linha

Fio sem fonte fica **indefinido (X)**; dois valores na mesma linha dão **conflito**. No Logisim os dois aparecem pela cor do fio (quadro 13).

4. Entradas, saídas, constantes e clock

pino de entrada

pino de saída

constante

$V_{CC} = 1$

$GND = 0$

clock

LED

Nível lógico: **0** = baixo (0V, falso), **1** = alto (V_{CC}). **Z** (alta impedância) e **X** (indefinido) não são níveis lógicos: são estados do fio.

6. Aritmética: somadores e ULA

meio somador

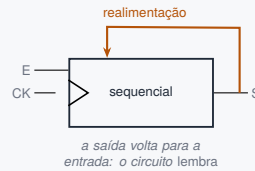
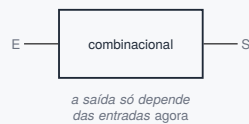
somador completo

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

unidade lógica e aritmética

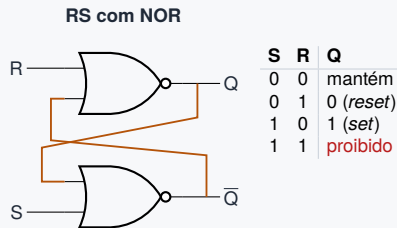
O V no topo é a marca da ULA: dois operandos por cima, resultado por baixo, código da operação pela lateral. *Flags* típicas: Z (zero), C (carry), N (negativo), V (overflow).

7. O que torna um circuito sequencial



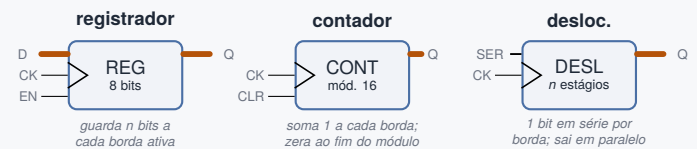
Só aqui aparecem o *clock* (diz *quando* a saída pode mudar) e as entradas assíncronas *Preset/Clear* (mudam a saída sem esperar o *clock*).

8. Latch — sensível a nível



Com o nível de habilitação ativo o *latch* segue a entrada — inclusive o ruído. Com NAND tudo se inverte: entradas $\bar{S}$ e $\bar{R}$, ativas em 0, e a combinação proibida passa a ser 00.

10. Registradores e contadores



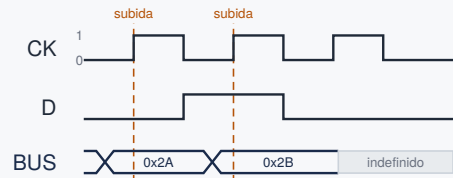
Azul claro = memória, em toda figura da disciplina. Fio fino = 1 bit (serial); barra grossa âmbar = barramento de n bits (paralelo) — a mesma convenção do quadro 9 em diante.

11. Memórias RAM e ROM



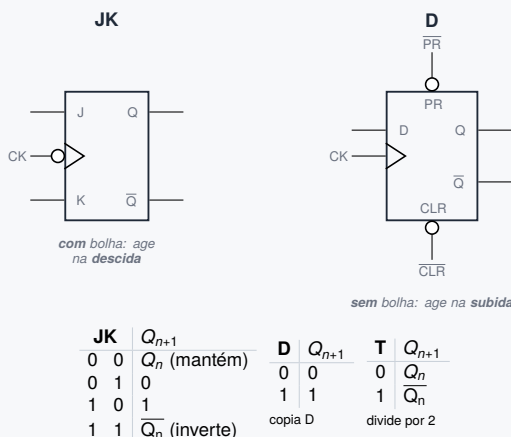
Capacidade = $2^{\text{bits de endereço}} \times \text{largura da palavra}$. $A_0 \dots A_9$ = endereço; $D_0 \dots D_7$ = dados (bidirecional na RAM). $R/\bar{W}$: 1 lê, 0 escreve; **ME** habilita a memória.

12. Cronogramas (timing diagrams)



O tempo corre para a direita; um degrau é uma transição. A tracejada marca a **borda ativa** — só ali a saída de um *flip-flop* muda. No barramento, as duas linhas cruzam a cada troca de valor.

9. Flip-flop — disparado por borda



A **cunha** ▸ no *clock* distingue *flip-flop* de *latch*. $\overline{PR}$ e $\overline{CLR}$ são **assíncronas** e não podem valer 0 juntas. T = JK com J = K; D = JK com K = $\bar{J}$.

13. Do símbolo ao Logisim — onde achar cada componente

AND, OR, NAND, NOR, XOR, XNOR → Gates · AND Gate, OR Gate, NAND Gate... *Number of Inputs, Data Bits, Gate Size, Negate* por entrada (é a bolha).
NOT, BUFFER, tri-state → Gates · NOT Gate, Buffer, Controlled Buffer/Inverter. *Data Bits, Control Line*.
Pino de entrada / de saída → Wiring · Pin. *Output?* (não = entrada), *Data Bits, Label*.
Constante, V_{CC} , GND → Wiring · Constant, Power, Ground. *Value, Data Bits*.
Clock → Wiring · Clock. *High/Low Duration*; simular com Ctrl+K ou Ctrl+T.
Túnel → Wiring · Tunnel. *Label* — mesmo rótulo, mesmo nó.
Splitter e barra de bits → Wiring · Splitter. *Fan Out, Bit Width In*, distribuição dos bits.
Pull-up/down, sonda de valor → Wiring · Pull Resistor (*Pull Direction*), Probe (*Radix*).
LED, botão, display → Input/Output · LED, Button, Hex Digit Display. *Cor, Active On High*.
MUX, DEMUX → Plexers · Multiplexer, Demultiplexer. *Select Bits, Data Bits, Include Enable?*

Cor do fio é diagnóstico, não decoração: cinza = sem valor, azul = indefinido (X), verde-escuro = 0, verde-claro = 1, preto = barramento, **vermelho** = conflito, **laranja** = largura incompatível — os dois últimos são erro de montagem.

Decodificador, codificador → Plexers · Decoder, Priority Encoder. O codificador do Logisim é **de prioridade**.
Somador de n bits, subtrator, comparador, deslocador → Arithmetic · Adder, Subtractor, Comparator, Shifter. O Adder já traz C_{in} e C_{out} .
Flip-flop SR, D, T, JK → Memory · S-R/D/T/J-K Flip-Flop. *Trigger: rising/falling edge; Preset e Clear* nos cantos.
Registrador, contador, deslocamento → Memory · Register, Counter, Shift Register. *Trigger, Maximum Value* (o módulo), *Number of Stages*.
RAM, ROM → Memory · RAM, ROM. *Address Bit Width, Data Bit Width*; conteúdo pelo *Poke Tool*.
Meio somador e somador completo → *não existem prontos* — montar com XOR + AND, como no quadro 6.
Latch RS e D → *não existem prontos* — montar com NAND/NOR (quadro 8); o Logisim só traz *flip-flops* de borda.
ULA → *não existe pronta* — montar com Adder + portas + Multiplexer escolhendo a operação.
Cronograma → menu *Simulate* · *Logging* escolhe os sinais e exporta; o Logisim Evolution tem *Chronogram*.