

Bits e Processadores

Lógica Sequencial e Memórias

"É mísera espécie de memória essa, que só funciona para trás."

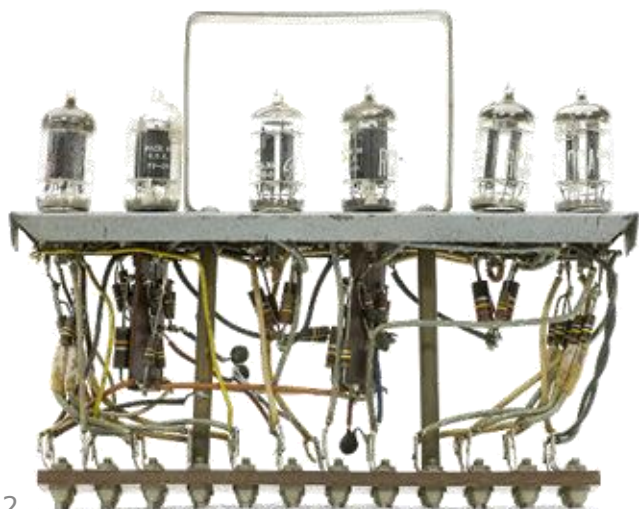
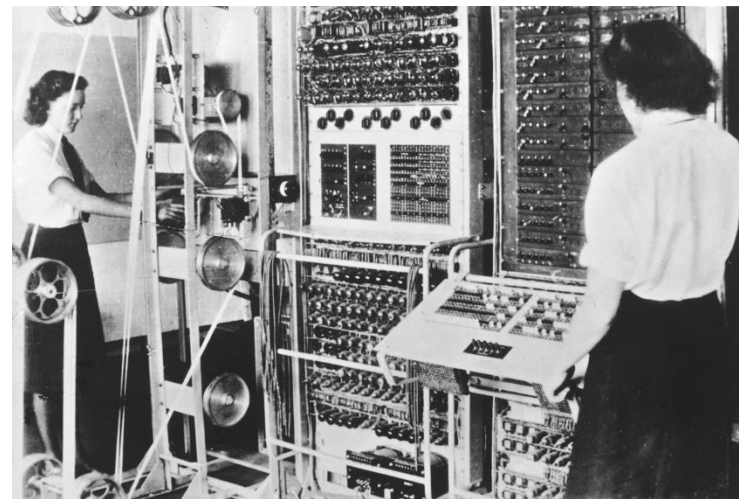
"It's a poor sort of memory that only works backward."

Lewis Carroll (1832–1898) , Romancista Americano

apud Nisan, N. & Schocken, S. 2005. Elements of Computing Systems

Colossus

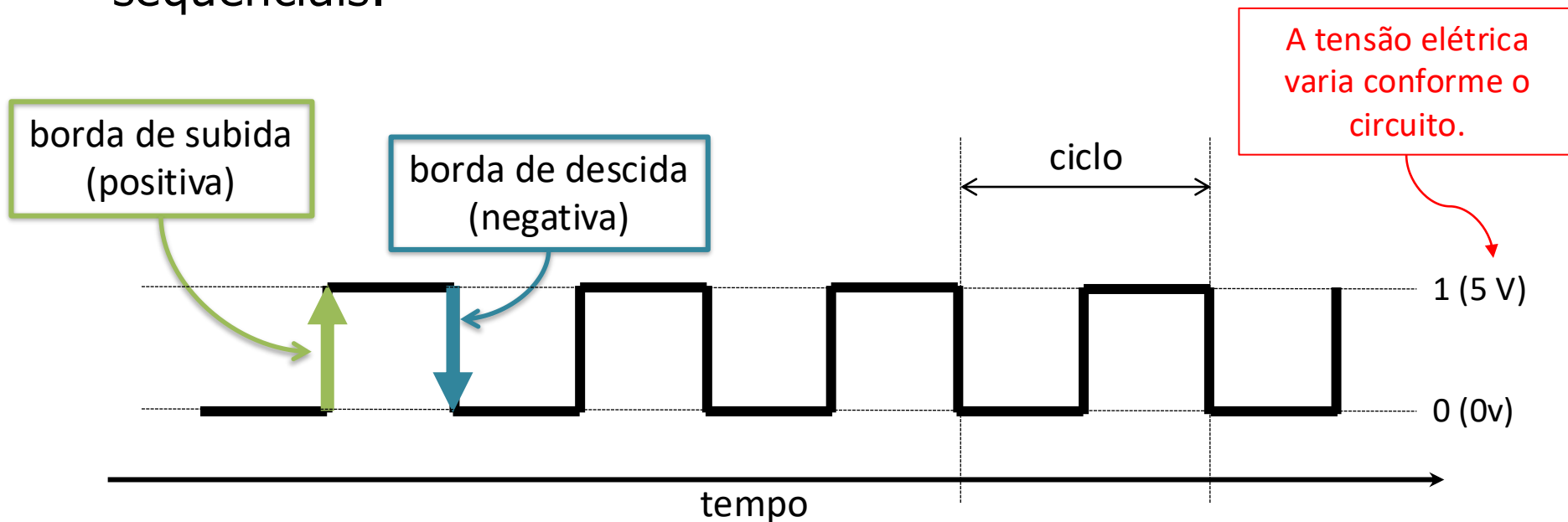
O Colossus foi um computador britânico de 1700 válvulas, feito em 1943 para decifrar códigos nazistas durante a Segunda Guerra Mundial. O computador foi desmontado e planos foram destruídos para manter o projeto em segredo.



Um dos sistemas de bits de memória do Colossus era armazenado em válvulas usando uma técnica desenvolvida por William Henry Eccles e Frank Wilfred Jordan em 1918, conhecida hoje como flip-flops.

Clock

Circuitos digitais síncronos utilizam sinais de clock para coordenar as mudanças de estado dos componentes sequenciais.



$$f = \frac{1}{T}$$

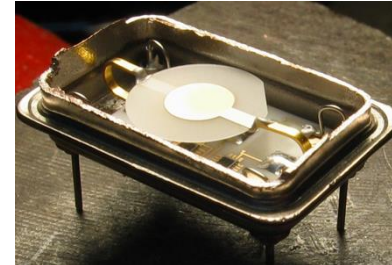
T é o período de tempo de um ciclo em segundos.
 f é a frequência em Hertz

Osciladores

Multivibrador Astável



Cristais de quartzo

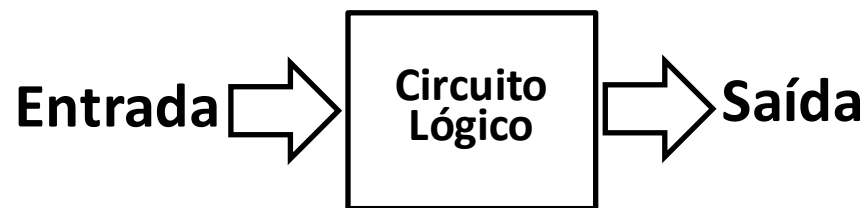


O Arduino Uno com um cristal de 16 MHz para controlar o ATmega328P

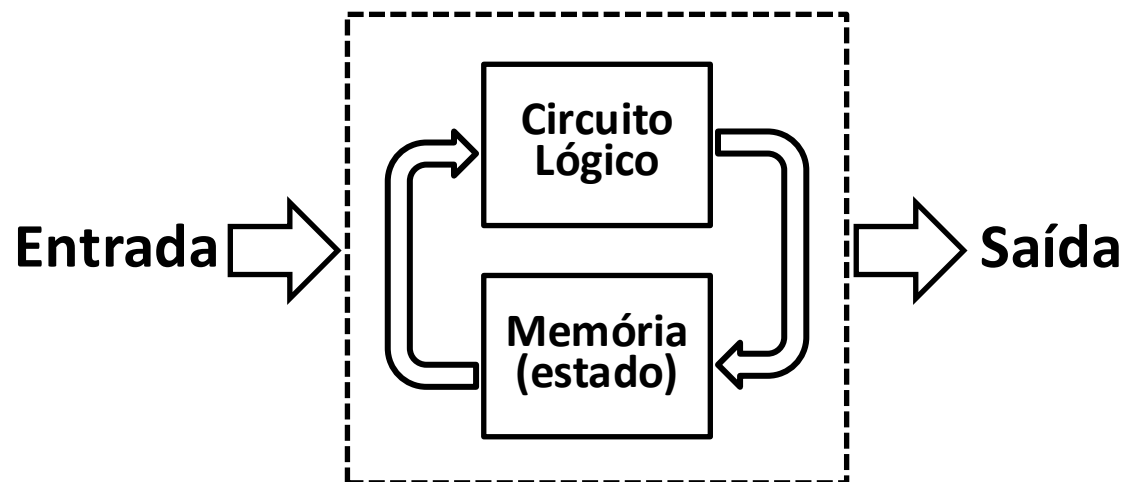


Lógica Combinacional e Sequencial

Na Lógica Combinacional a função dos circuitos depende somente do estado das entradas naquele momento.

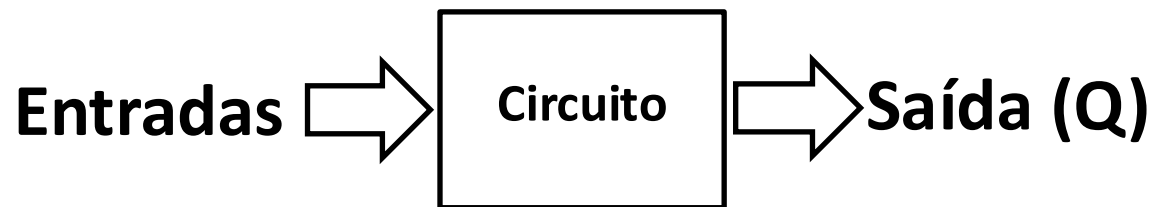


Na Lógica Sequencial a função dos circuitos depende de estados anteriores, ou seja, os circuitos têm alguma memória.



Circuitos Biestáveis

Um circuito biestável possui dois estados estáveis e pode armazenar um bit. Seu estado somente muda quando suas entradas e, quando aplicável, o sinal de habilitação ou clock determinam uma transição.

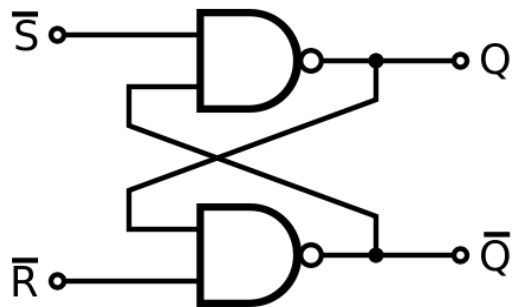
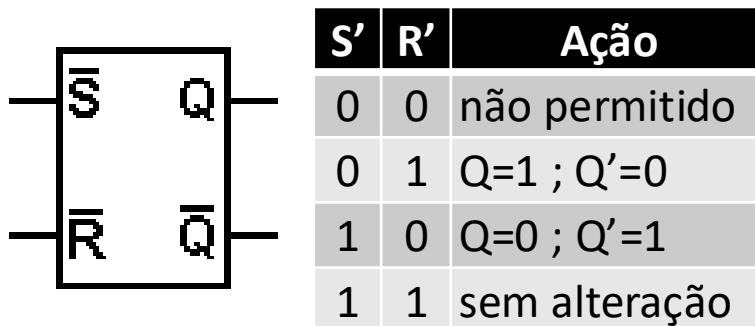


$$Q(t+1)=F(Q(t),entradas)$$

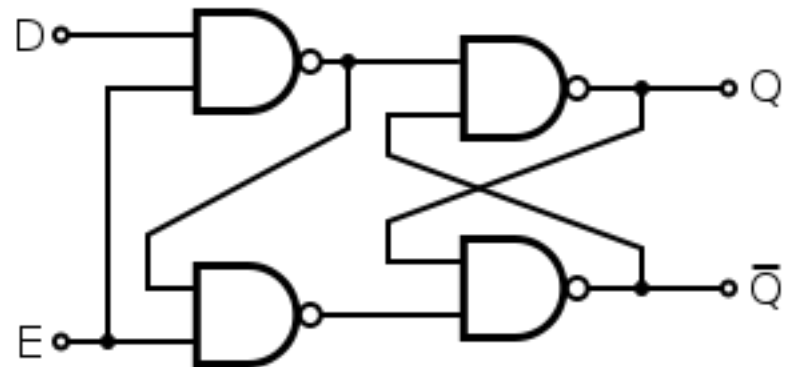
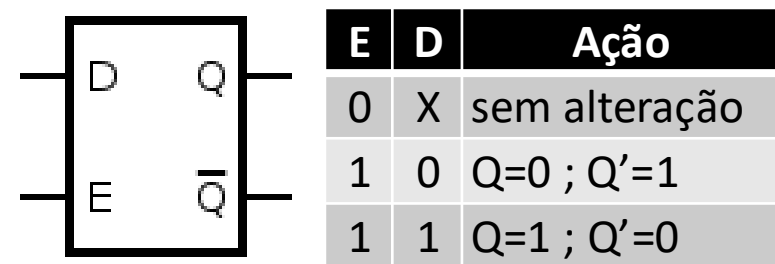
O próximo estado é uma função do estado atual e das entradas.

Tipos de Circuitos Biestáveis

Os tipos comuns são o SR ("Set-Reset") e o D ("data").



Latch de Set-Reset ativado com sinais em nível baixo.

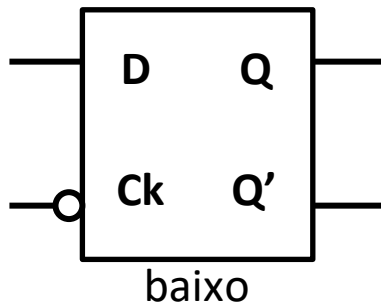
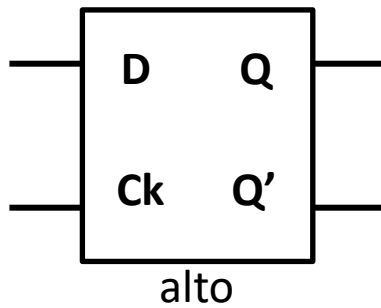


Latch de tipo D ativado com sinais em nível alto.

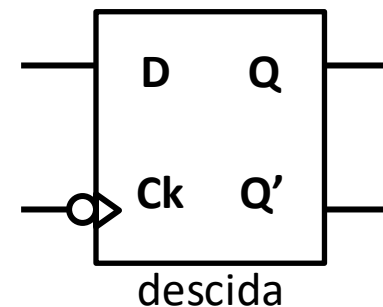
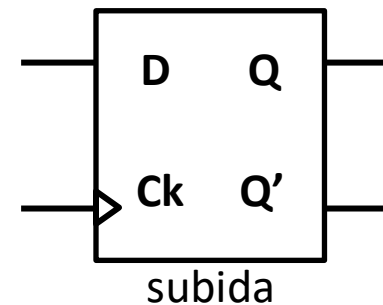
a condição $\bar{S}=\bar{R}=0$ é proibida/indeterminada!

Flip-flops e Latches

**Sensível ao nível
(latch)**

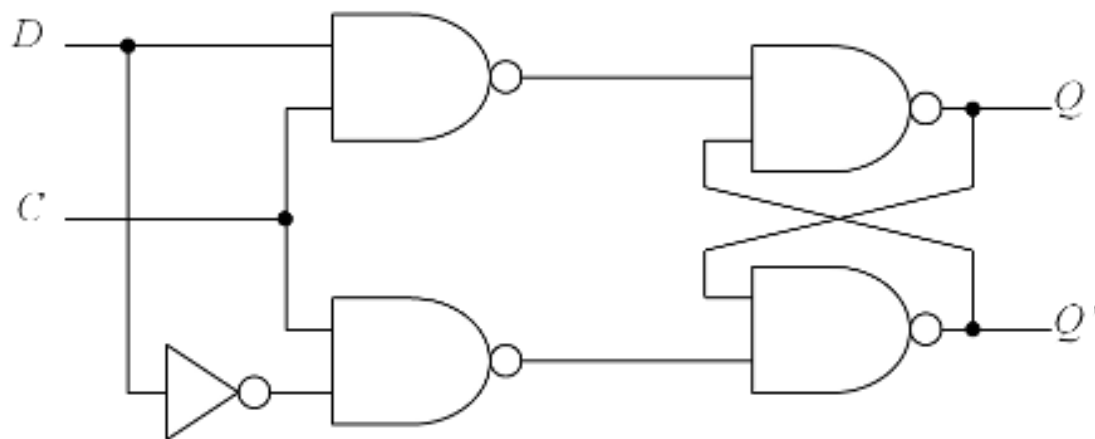


**Sensível à borda
(flip-flop)**

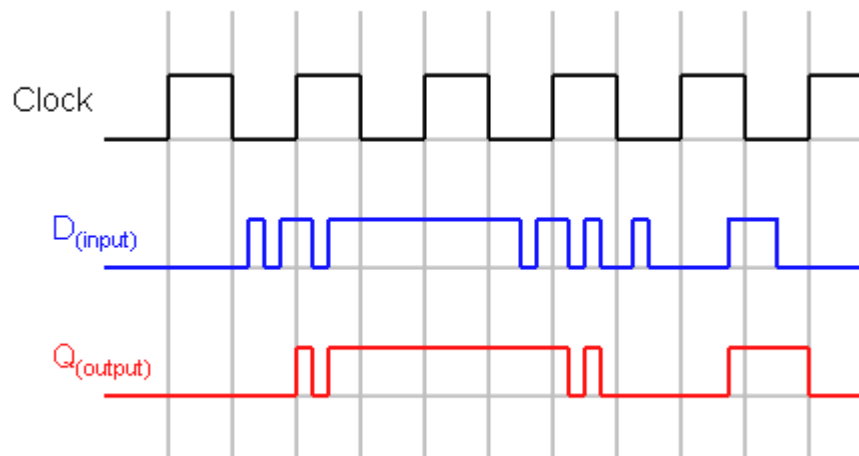


Latch tipo D

Sinais de clock podem ser usados para os *latches*.

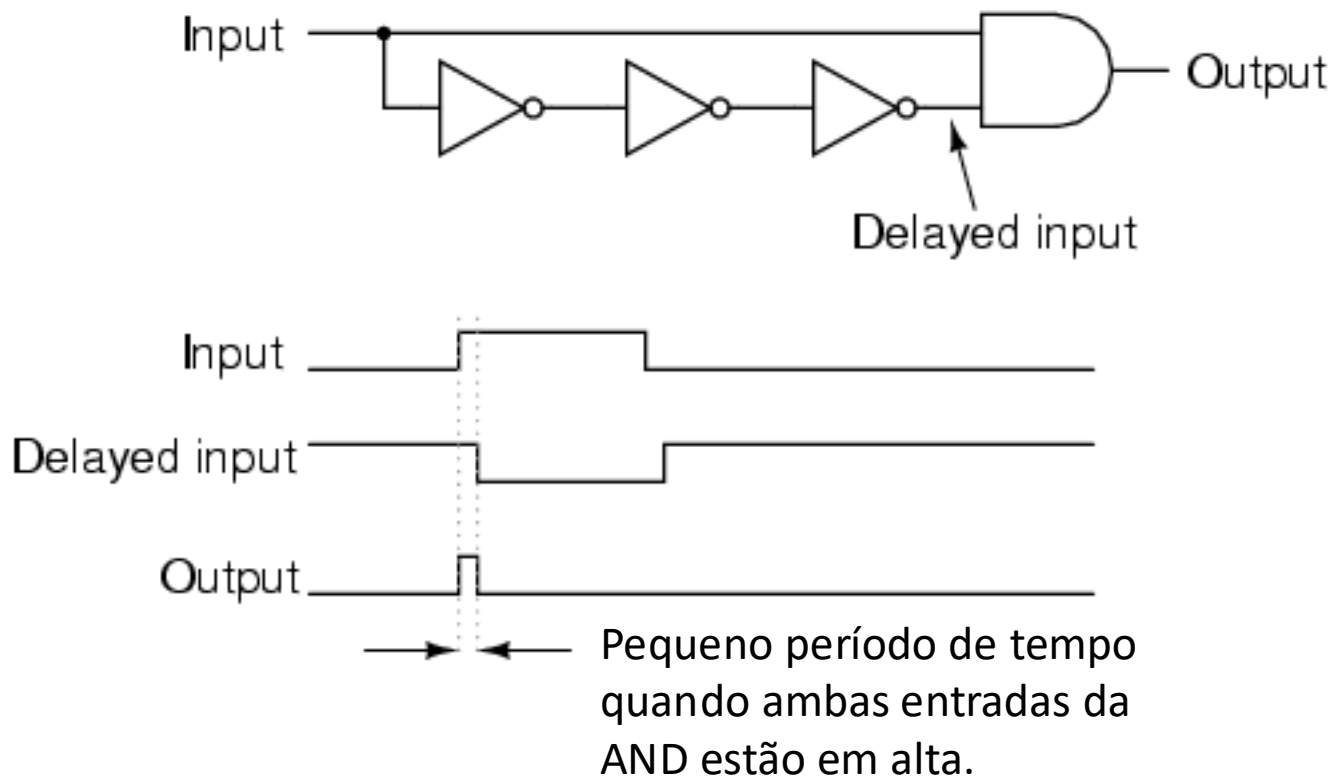


D	Clk	Ação
X	0	sem alteração
0	1	$Q=0$; $Q'=1$
1	1	$Q=1$; $Q'=0$



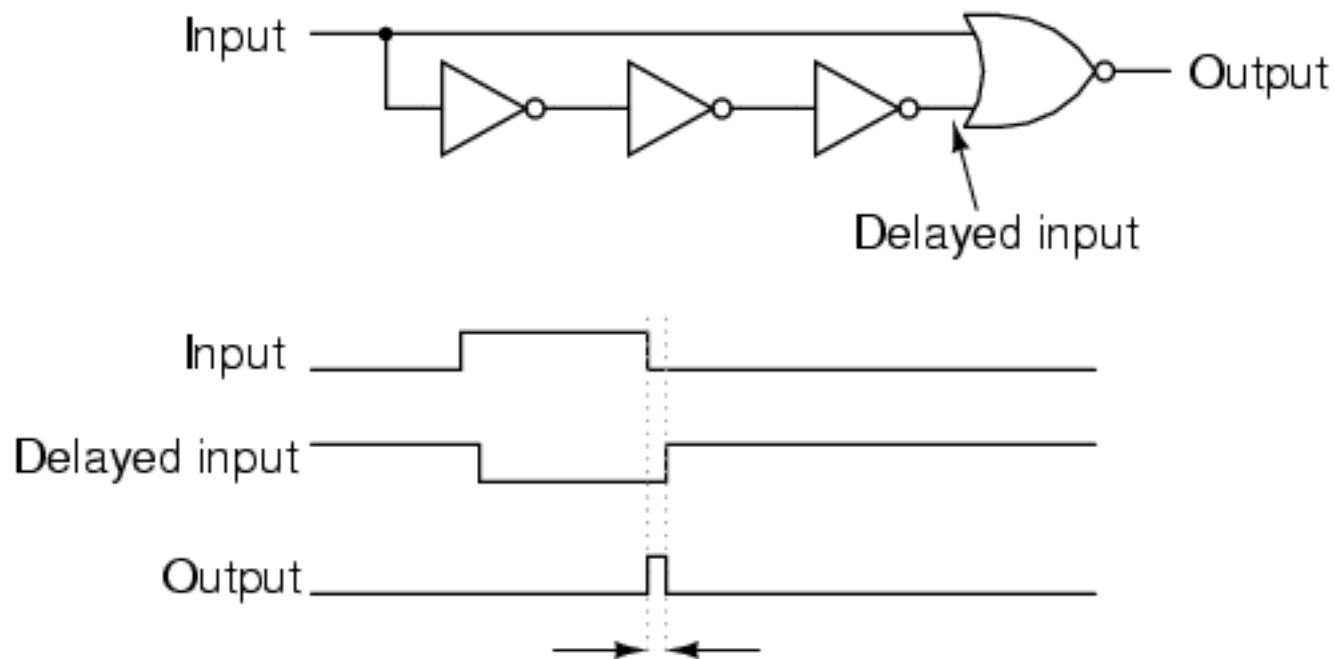
Acionamento por pulso (subida)

Circuitos de disparo permitem gerar um pulso curto.



Acionamento por pulso (descida)

Circuitos de disparo permitem gerar um pulso curto.

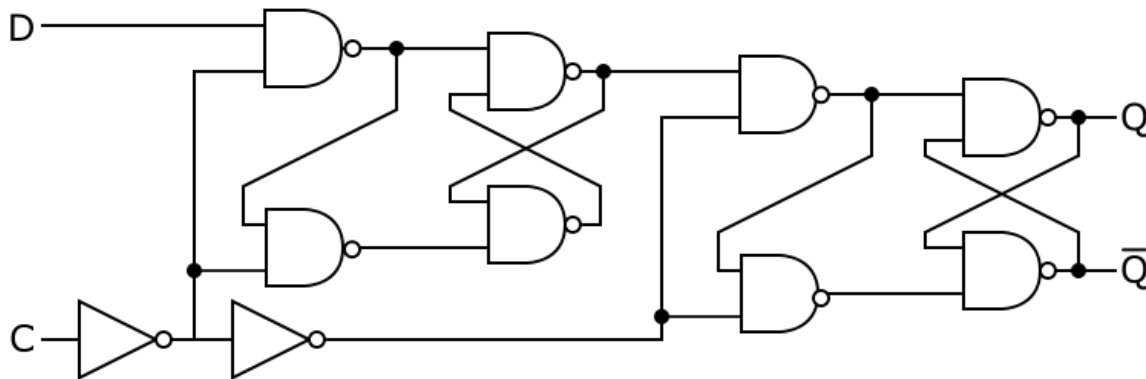


Pequeno período de tempo
durante o qual ambas as entradas
da porta NOR estão em nível baixo.

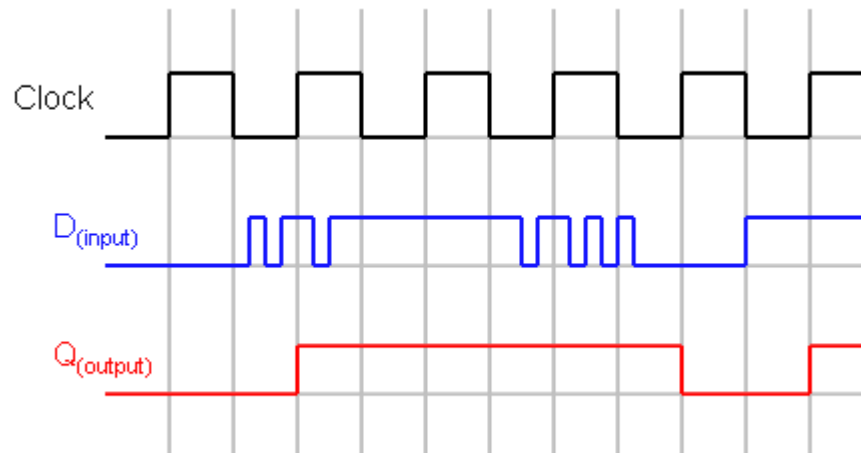
Esses circuitos são didáticos, mas não são bons para gerar clocks robustos!

Flip Flop

Flip flop tipo D ou DFF (Data Flip Flop), usando implementação master-slave.



D	C	Ação
X	0	sem alteração
X	1	sem alteração
1	↑	Q=1 ; Q'=0
0	↑	Q=0 ; Q'=1



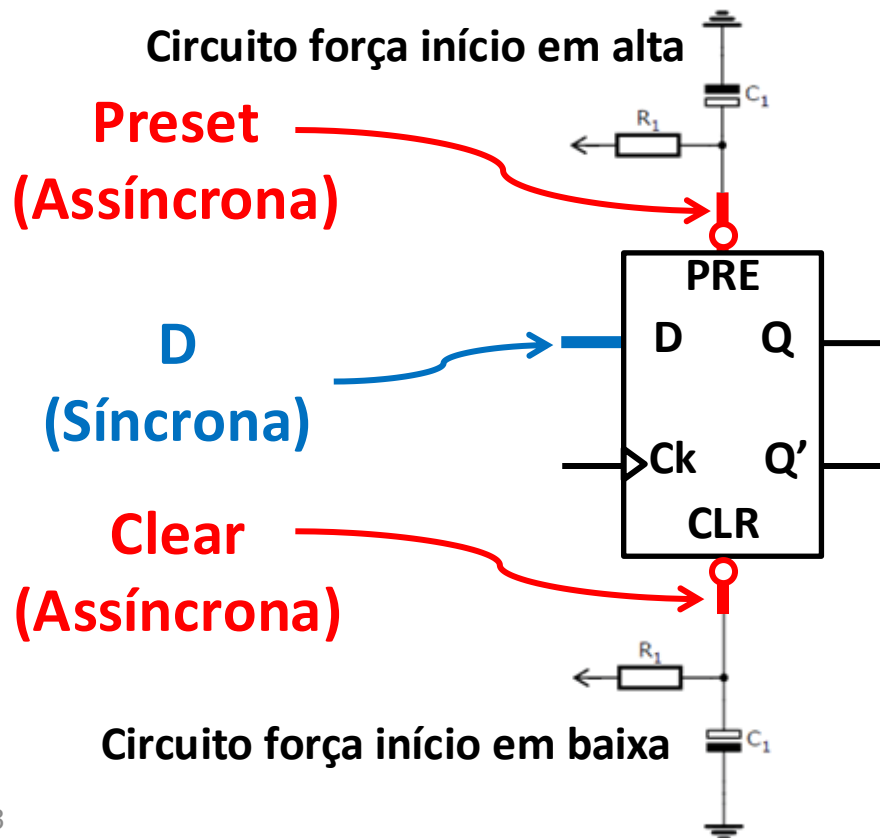
Somente sensível a
borda de subida!

Entradas síncronas e assíncronas

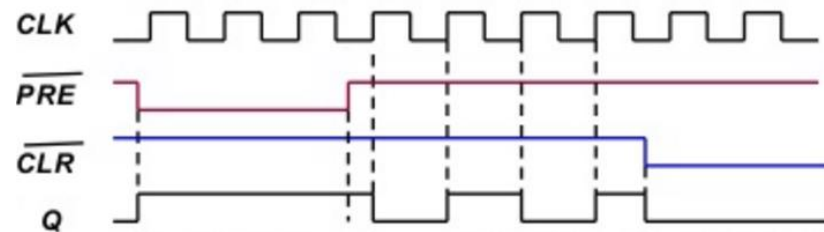
Componentes sequenciais podem ter entradas:

Síncronas: só alteram o estado do chip com os pulsos de clock

Assíncronas: alteram o estado do chip independentemente do clock



$\overline{PRE}$ e $\overline{CLR}$ são ativos em nível baixo

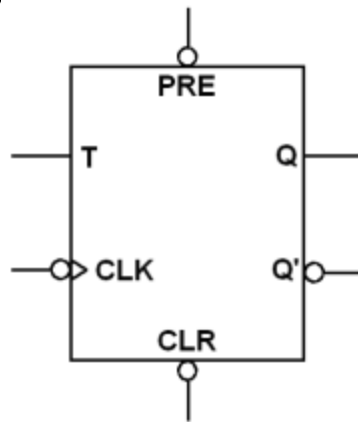


ativar preset e clear simultaneamente
costuma ser uma condição proibida!

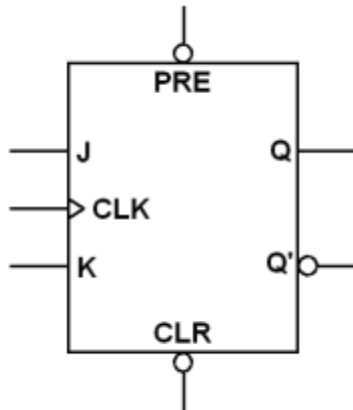
Flip Flop

Outros tipos de flip flops comuns são:

Flip-flop T (toggle)



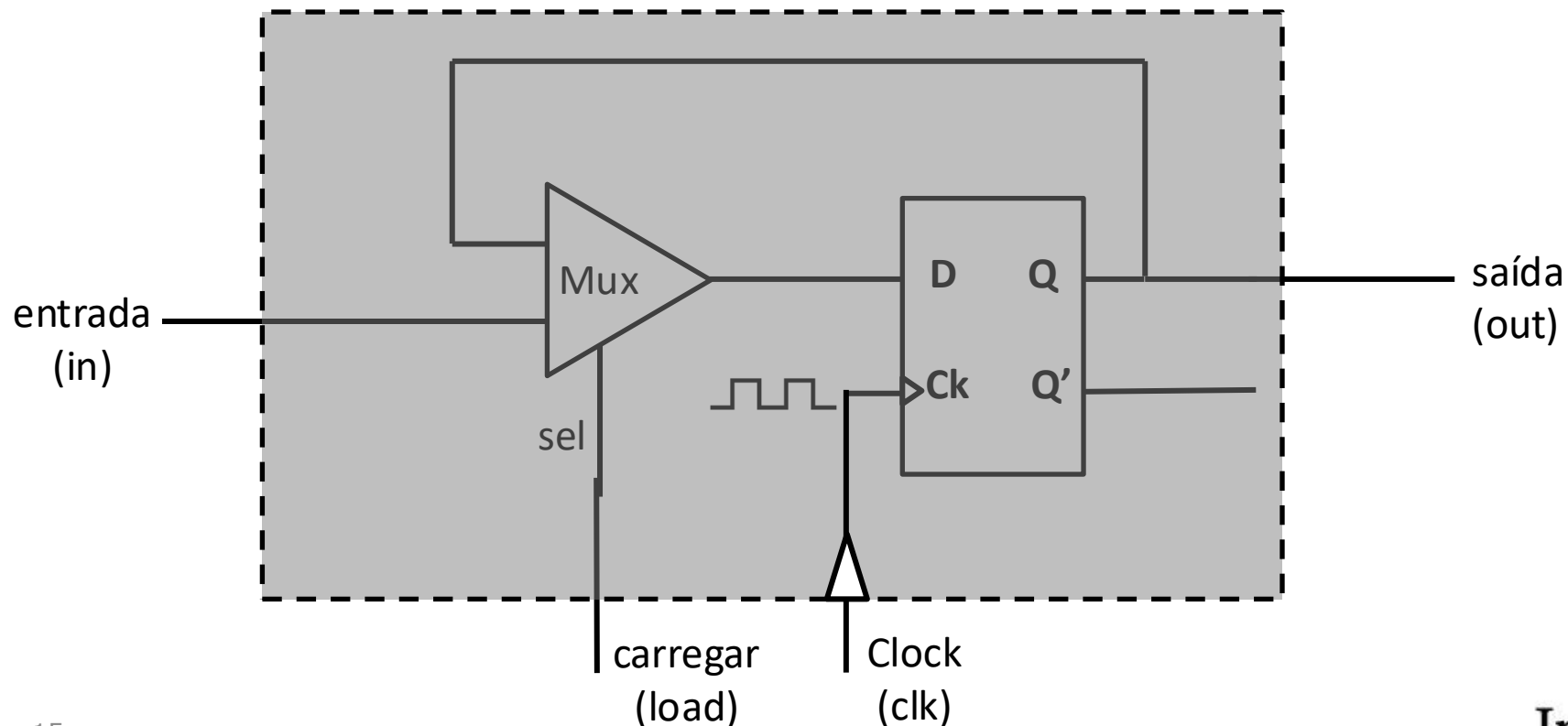
Flip-flop JK



Lendo e armazenando valores

Uma célula de memória.

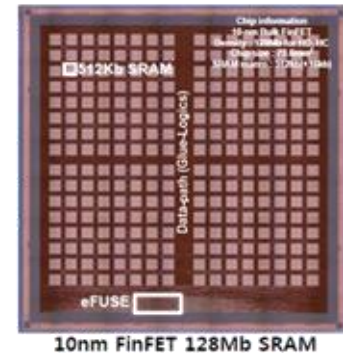
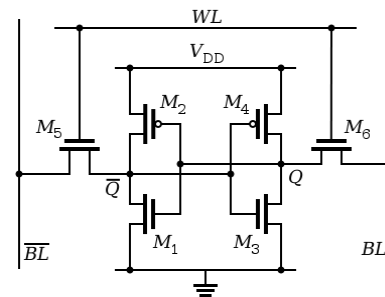
$$D_{FF} = \begin{cases} Q, & load = 0 \\ in & load = 1 \end{cases}$$



Memórias RAM Atuais

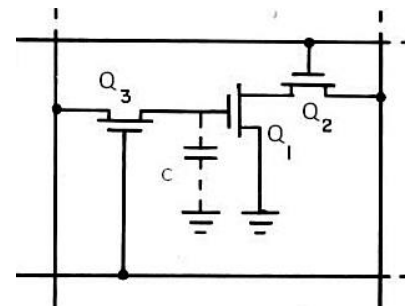
SRAM (Static Random-Access Memory)

Armazena cada bit em uma célula biestável, normalmente implementada com seis transistores. É rápida e não precisa de *refresh* enquanto estiver alimentada, sendo muito usada em caches e pequenas memórias internas.

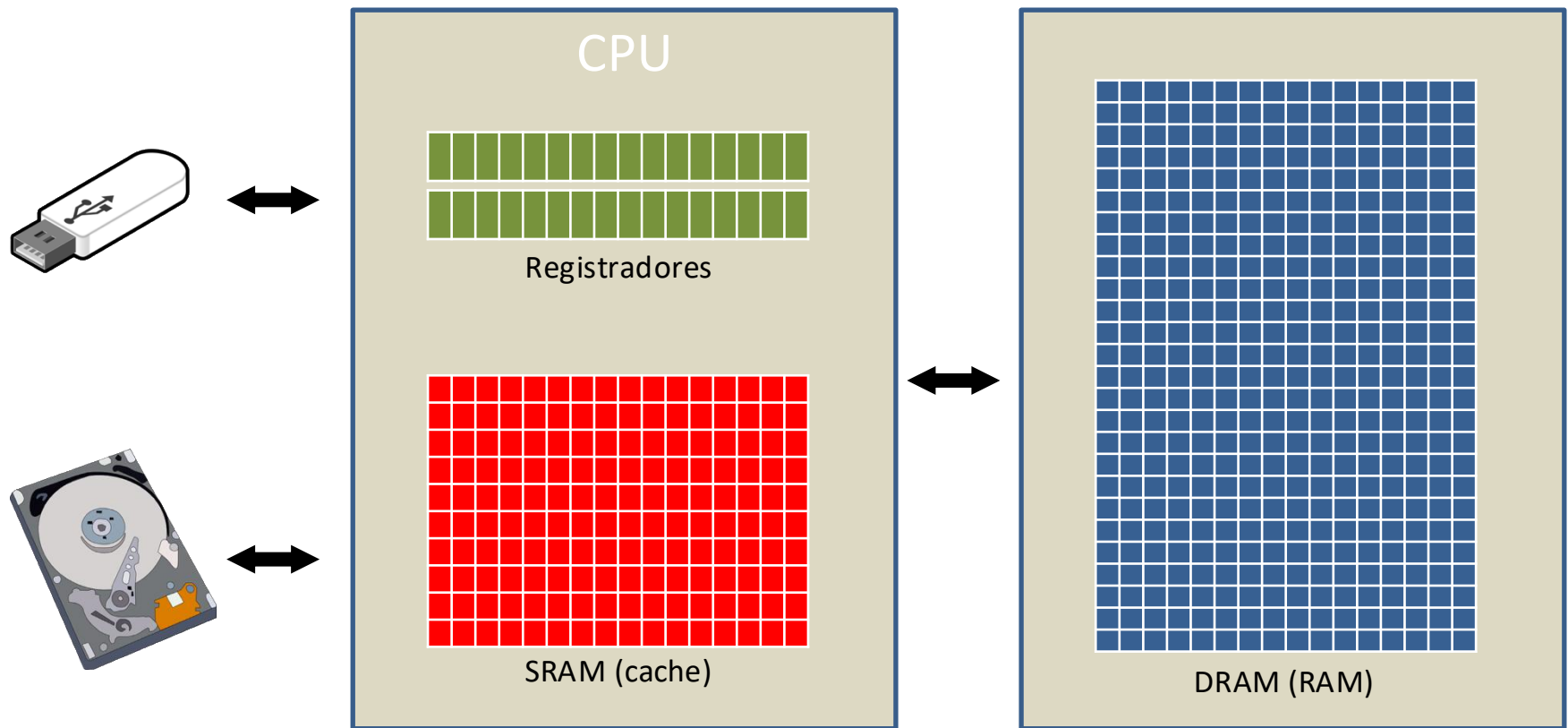


DRAM (Dynamic Random-Access Memory)

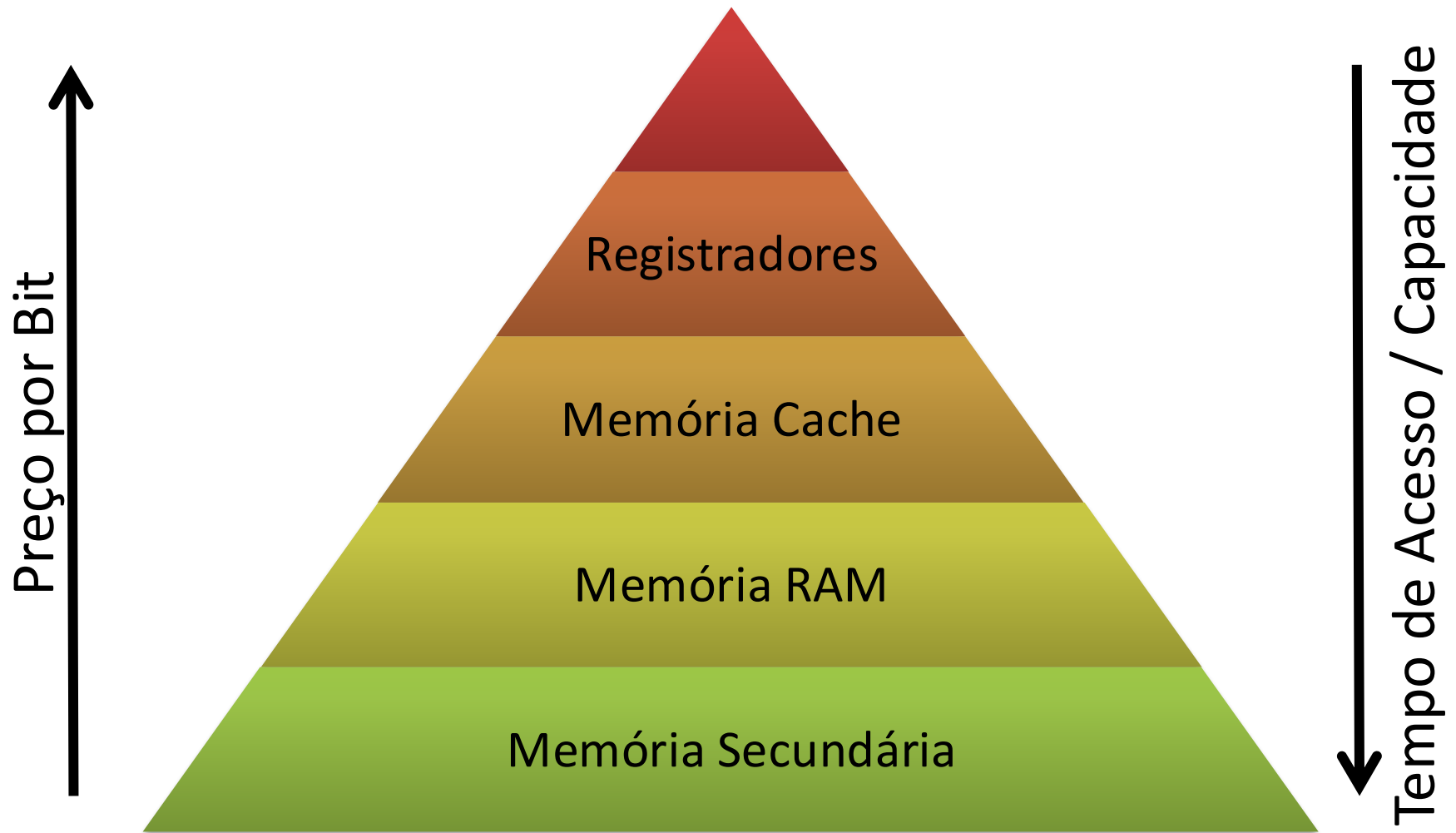
Armazena cada bit como carga em um capacitor. Como a carga se dissipa, as células precisam ser periodicamente atualizadas (refresh). Sua maior densidade e menor custo por bit favorecem seu uso como memória principal.



Hierarquia de Memórias

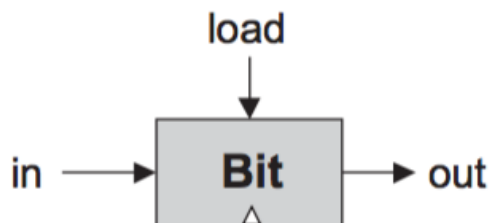


Hierarquia de Memória



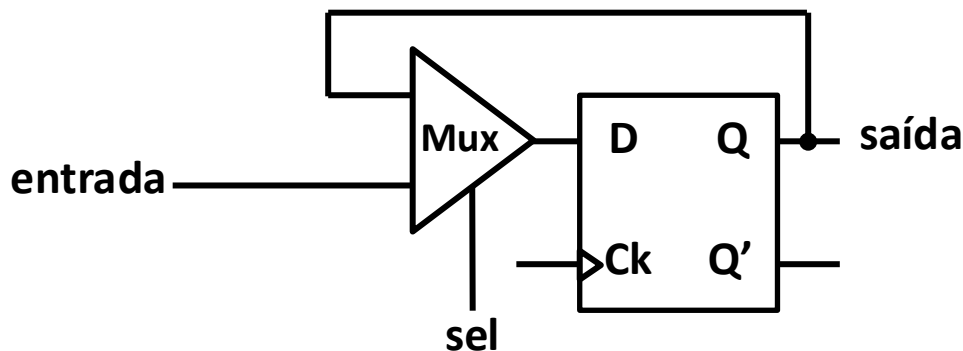
Mantendo um bit

Um *binary cell* é um elemento que mantém a memória de um bit e pode carregar um novo valor:



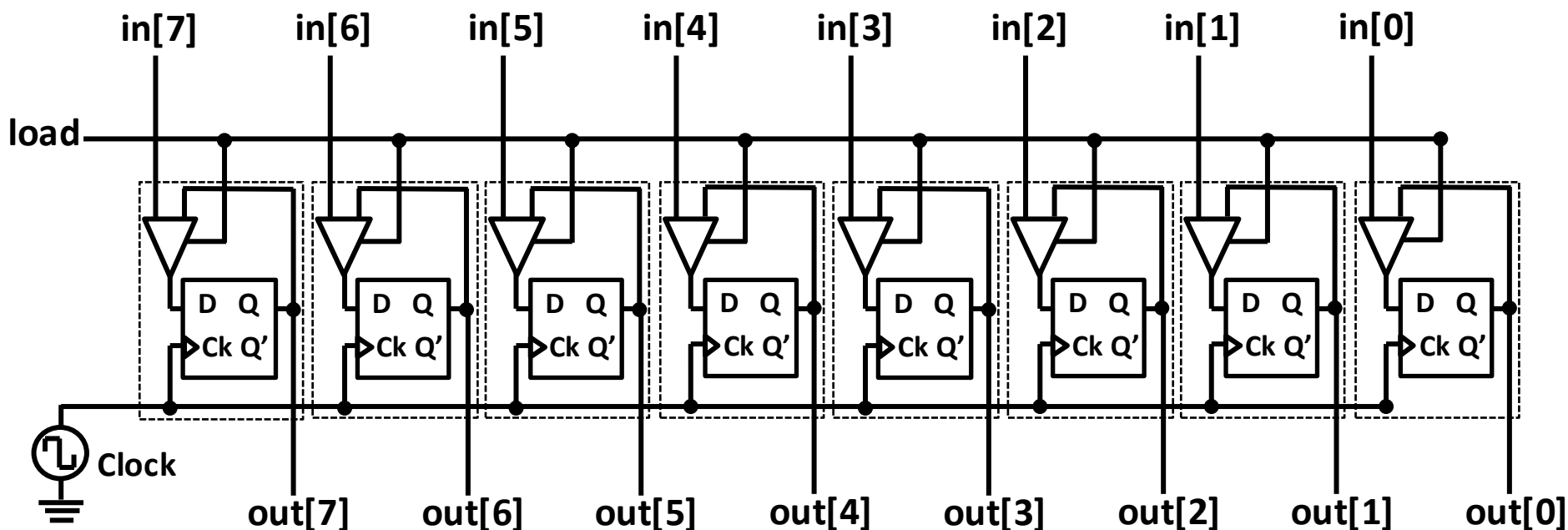
if load($t-1$) then out(t) = in($t-1$)
else out(t) = out($t-1$)

Uma forma de implementar o *binary cell* é realimentar um *flip-flop* com sua saída, ou seja, colocar o sinal da saída na entrada. Para controlar o carregamento, pode-se usar um multiplexador.



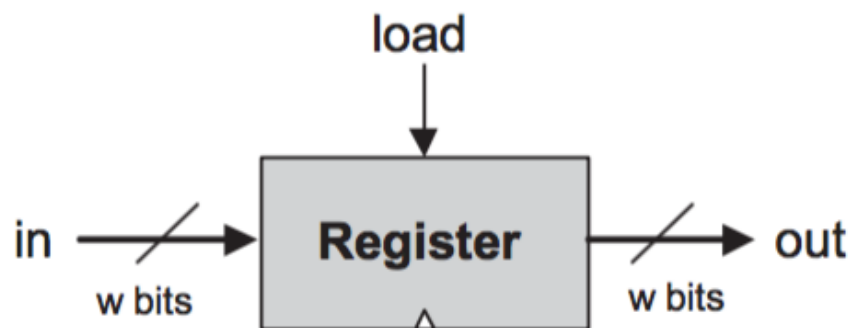
Blocos de Bits

Combinando várias células binárias, podemos construir um registrador de vários bits, como 8, 16 ou 32 bits. A quantidade de bits armazenados é chamada de largura ou comprimento da palavra (word length).



Registrador

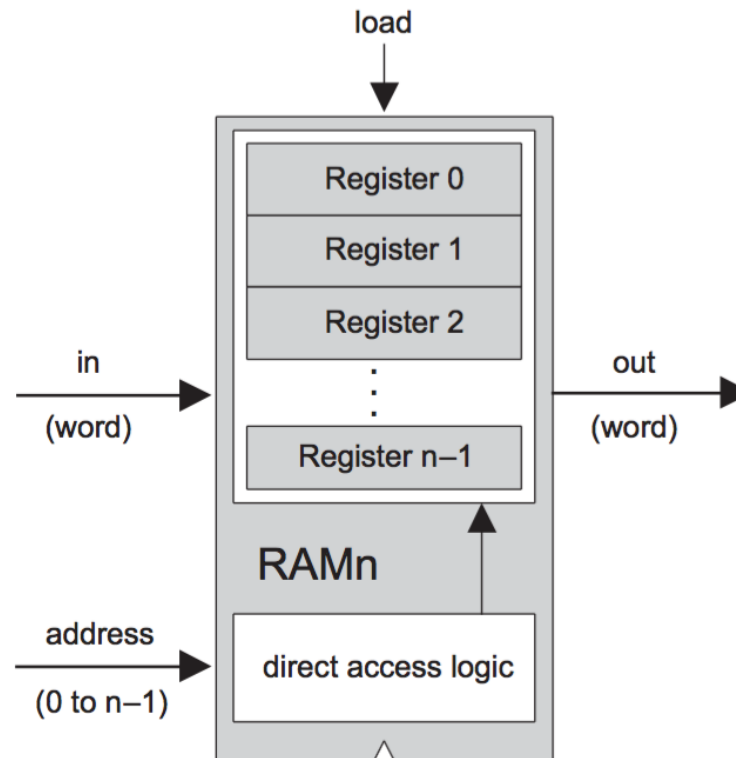
Registradores são elementos de armazenamento rápidos, internos ao processador, usados para guardar operandos, endereços, resultados e informações de controle. A largura dos registradores de propósito geral é uma das características usadas para classificar uma arquitetura como sendo de 8, 16, 32 ou 64 bits.



Dizemos que o estado do registrador é o valor nele armazenado!

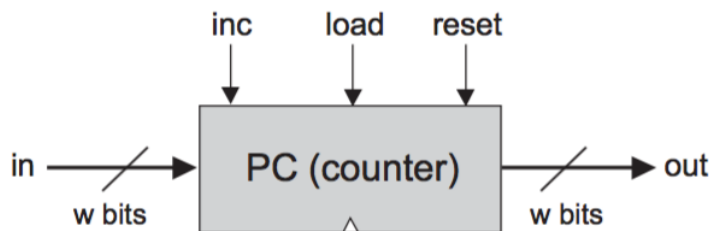
Memória RAM (Random Access Memory)

Juntando vários “registradores” é possível criar uma memória de qualquer tamanho. Um sistema de controle de endereçamento é necessário para se possam armazenar e recuperar dados em qualquer posição de memória.

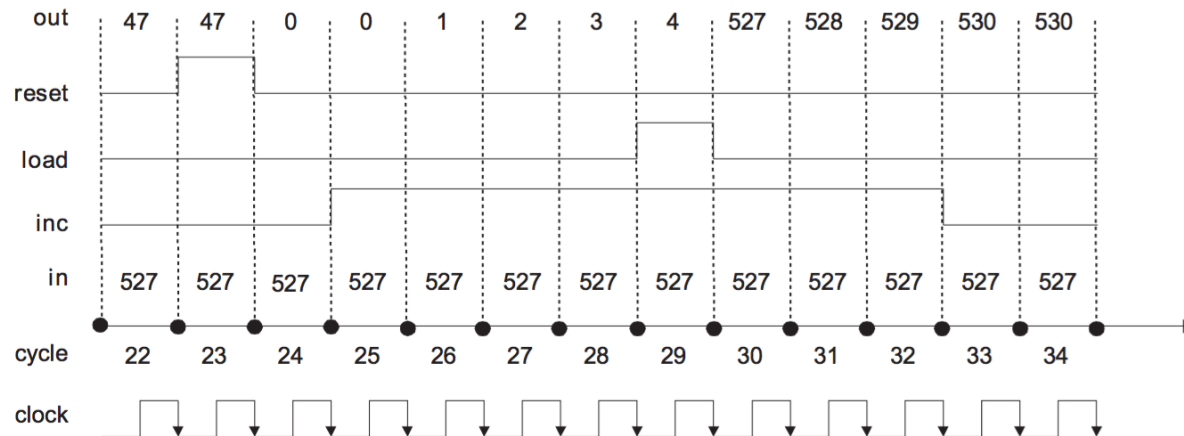


Program Counter (PC)

O *Program Counter* é um registrador que contém o endereço da instrução a ser executada. Em execução sequencial ele é incrementado. Saltos, chamadas, retornos, exceções e reset podem carregar outro endereço.

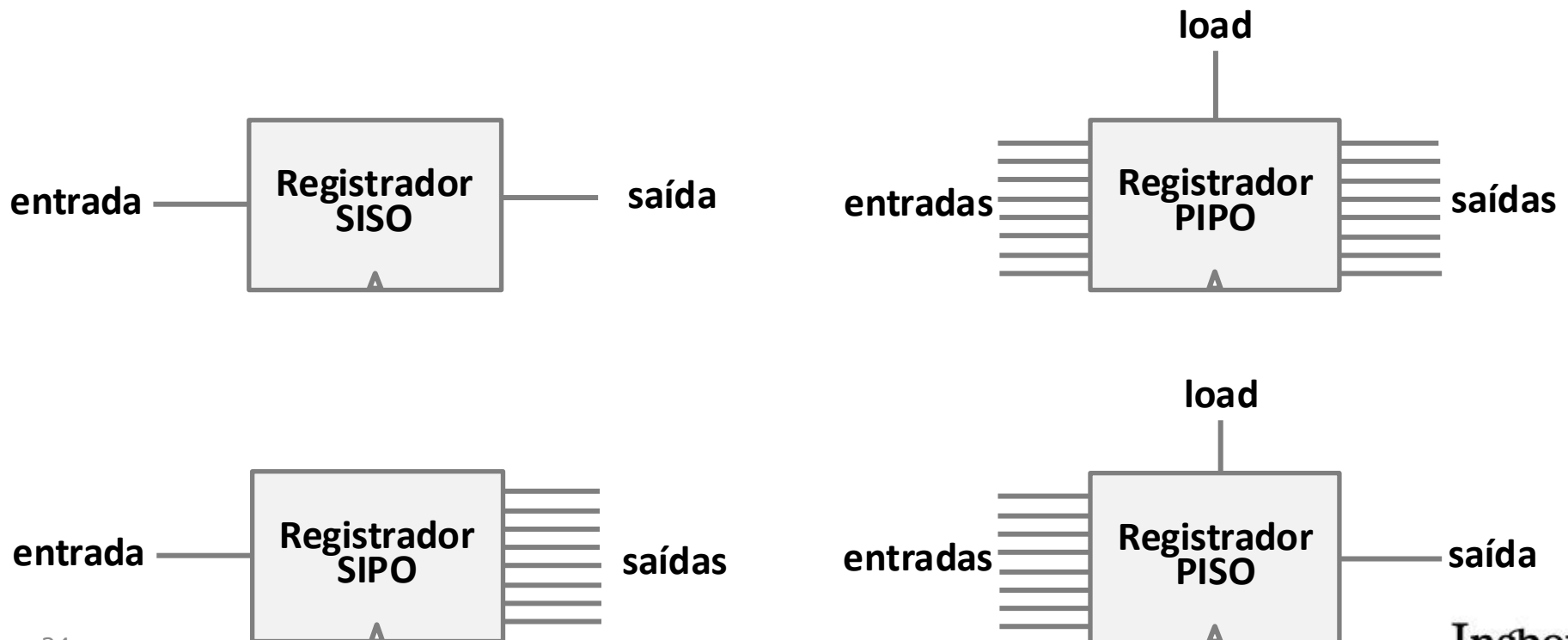


$$PC^+ = \begin{cases} 0, & \text{reset} = 1 \\ in, & \text{load} = 1 \\ PC + 1, & \text{inc} = 1 \\ PC, & \text{caso contrário} \end{cases}$$



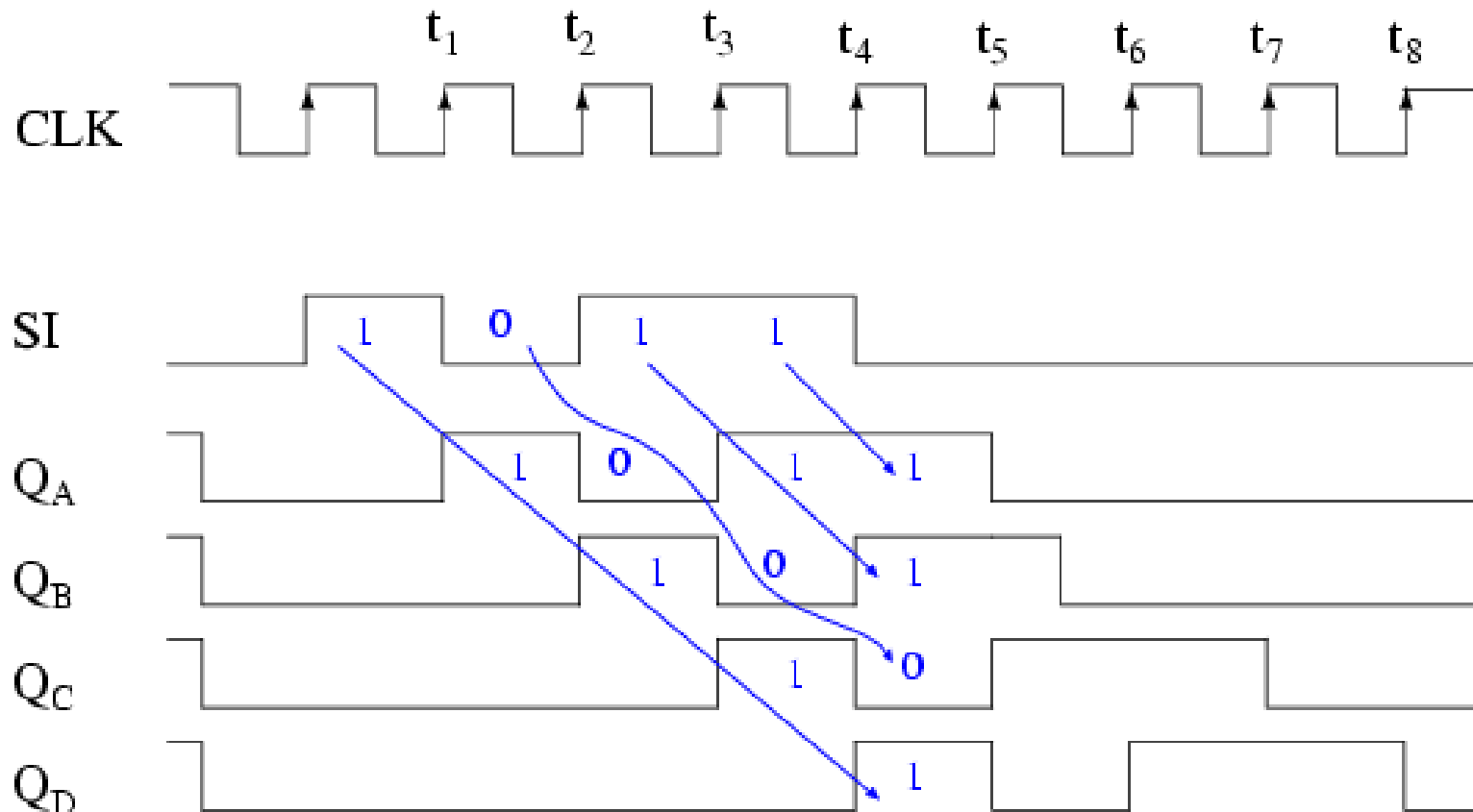
Registadores Seriais e Paralelos

Registadores podem receber e fornecer dados em formato serial ou paralelo. Registradores de deslocamento movem os bits entre seus estágios, permitindo conversões entre esses formatos.



Exemplo de SIPO

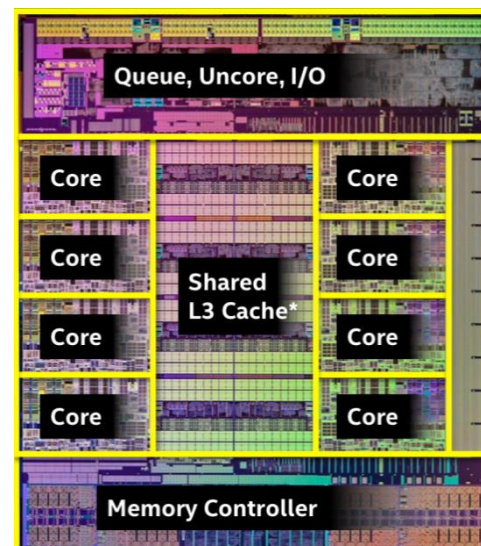
Serial In Parallel Out



O uso de registradores

- Em arquiteturas load-store, as operações aritméticas e lógicas atuam sobre registradores;
- Evitam acessos frequentes aos níveis mais lentos da hierarquia de memória.
- O acesso a registradores costuma ter baixa latência e é integrado ao pipeline do processador.

Caches podem ser usados para uma solução meio-termo:



Intel Core i7-5960X
Proc. de oito núcleos

Insper

www.insper.edu.br