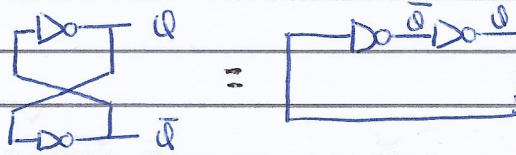


### III Circuitos sequenciais

memórias são circuitos em que a saída depende do valor atual e do valor passado das entradas

circuito bi-estável



tem dois estados estáveis

armazena valor de Q

se Q for 0, continuará 0 p/ sempre

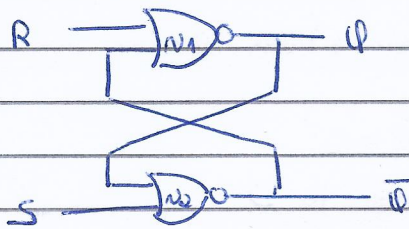
se Q for 1, continuará 1 p/ sempre

n estados estáveis armazenam  $\log_2 n$  bits de informações

desvantagens não tem aplicações prática pois não há controle sobre inputs

Latch SR circuito bi-estável (1 NOR em vez de 2)

2 entradas: SET e RESET

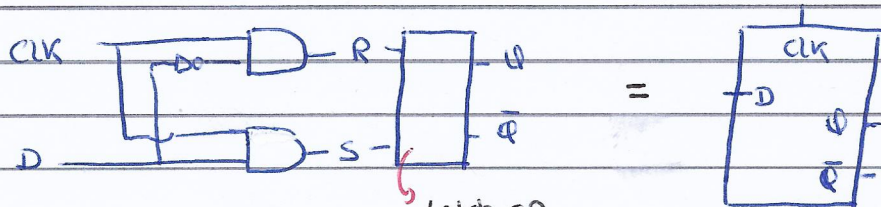


S	R	Q
✓ 0	1	0
✓ 1	0	1
✓ 0	0	estado anterior
X 1	1	Q = Q-bar X estado inválido

desvantagem estado inválido  $S1 \wedge R1 \rightarrow Q = \bar{Q}$   
confunde quando o como

Latch D 2 entradas: D e CLOCK

- 1 Q = D
- 0 Q = Q previous



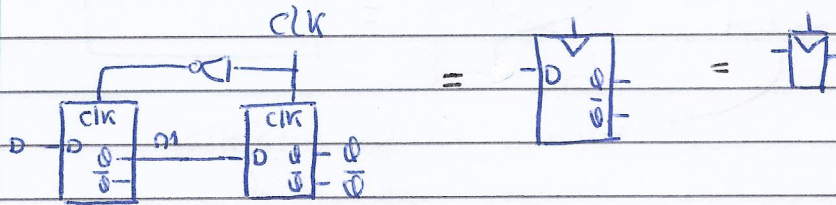
Latch SR

vantagem evita estado inválido

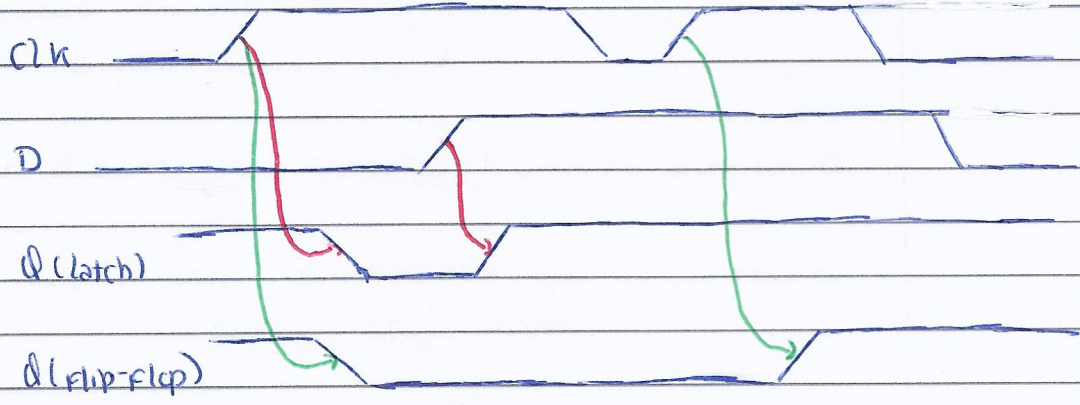
flip-flop D

como Latch D mas escrita ocorre apenas quando o relógio transita de 0 para 1

edge-triggered ativo na transição  
há apenas um instante em que escrita é efetuada

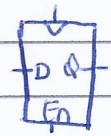


latch D vs  
flip-flop D



variações do  
flip-flop D

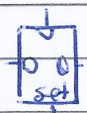
enable | controle qnd há escrita de dados  
 $En = 0 \rightarrow \bar{n}$  há escrita  
mesmo q haja clk



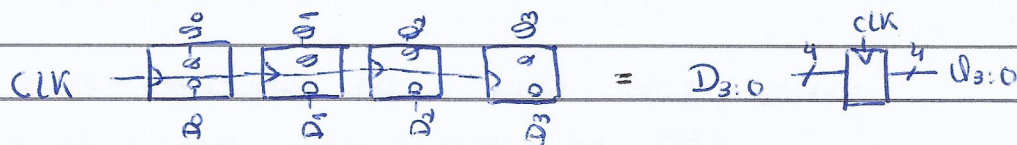
reset |  $Reset = 1 \rightarrow Q = 0$   
independentemente de D



set |  $Set = 1 \rightarrow Q = 1$   
independentemente de D



registros um flip-flop D armazena 1 bit  
 n flip-flops D armazenam n bits



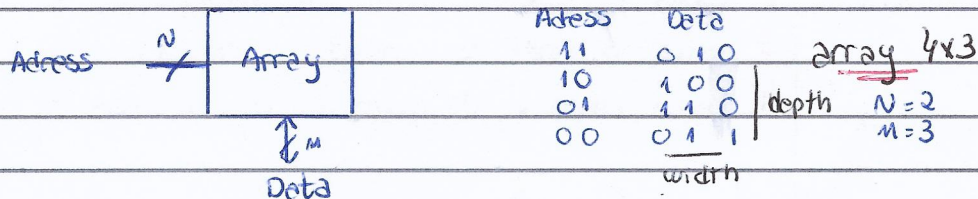
memórias armazenam de forma eficiente grandes quantidades de dados

tipos	DRAM	Dynamic Random Access Memory	} voláteis
	SRAM	static Random Access Memory	
	ROM	Read only Memory	↳ não voláteis

RAM volátil perde informações quando desligada  
 pode ser lida e escrita rapidamente  
 pode acessar-se com igual facilidade a qdq posições de memória (ao contrário das de acesso sequencial)

ROM não volátil mantém informações quando desligada  
 acesso para leitura é rápido  
 para escrita é impossível ou lento  
 obs antigamente com programadas em fábrica.  
 hoje com a tecnologia flash memory já não é verdade

arrays de memória a memória é organizada num array bidimensional de células de memória. Esta lê ou escreve os conteúdos de uma das linhas do array, linha esta que é especificada por um endereço. O valor escrito ou lido é chamado "data".



Um array com  $N$ -bit endereços e  $M$ -bit data tem  $2^N$  linhas e  $M$  colunas. Cada linha de "data" é chamada de "word".

Um array contém assim  $2^N \times M$  palavras (words).

tamanho  $\rightarrow$  depth (linhas)  $\times$  width (colunas)

espaço de endereçamento | número de endereços possíveis ( $2^N$ )  
 endereçabilidade | número de bits em cada endereço ( $M$ )

endereço | número único contado sequencialmente a partir do zero que identifica cada posição de memória

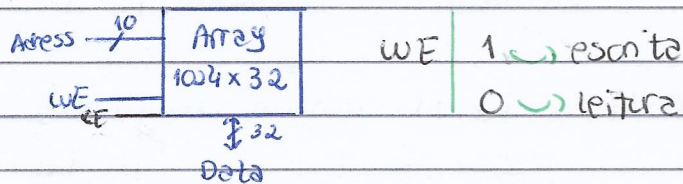
espaço de endereçamento | gama total de endereços que o CPU consegue referenciar (por ex 16 bits:  $[0, 2^{16}-1]$ )

aplicações | memória com  $2^k \times m$  bits

- endereço identificado com  $k$  bits
- espaço de endereçamento entre  $[0, 2^k-1]$
- conteúdo são palavras de  $m$  bits

escrita vs leitura

distinguem-se as operações através do write Enable (WE)



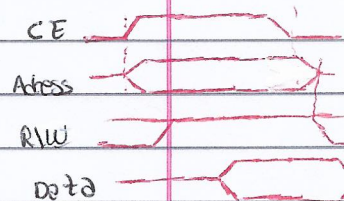
escrita valor das linhas Data é guardado no endereço Address  
 leitura valor guardado no endereço Address é colocado nas linhas Data

CE chip enable, ativa array de memória (seleciona)

ciclo de acesso à memória

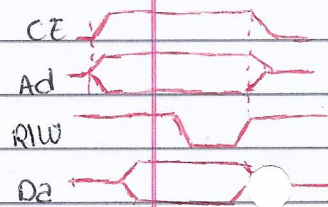
leitura (load)

1. ativar o sinal de CE
2. colocar o endereço a ler no barramento de endereços
3. envia o sinal de read ( $WE = 0$ )
4. lê o conteúdo da memória no barramento de dados



escrita (store)

1. ativar o sinal de CE
2. colocar o endereço a ser escrito no barramento de end.
3. colocar o valor a escrever no barramento de dados
4. ativar o sinal de write



celulas de memoria | unidade basica capaz de armazenar um bit que compoe os array de memoria

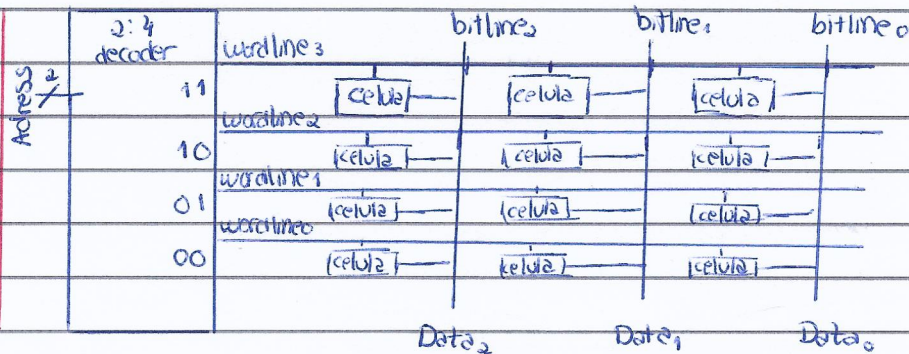
interfaces | wordline | para saber quando e endereçada  
comum a todos os elementos de uma palavra  
seleciona a linha a ser acessada (como enable)  
só pode haver uma ativa de cada vez

bitline | wordline ativa  
o conteúdo da célula passa para a bitline  
wordline inativa  
célula não interfere (alta impedância ao instate)

processos | leitura | 1° bitline em alta impedância  
2° wordline ativada  
3° bitline toma valor de célula

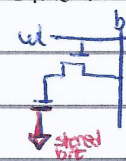
escrita | 1° bitline "forçamente induzido" ao valor a escrever  
2° wordline ativada  
3° bitline "sobrepõem-se" ao conteúdo da célula

organização da memória  
array 4x3



memória

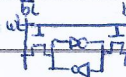
dinâmica



bit guardado como uma carga num condensador  
este sofre perdas de carga, pelo que armazenamento é temporal  
leitura é destrutiva

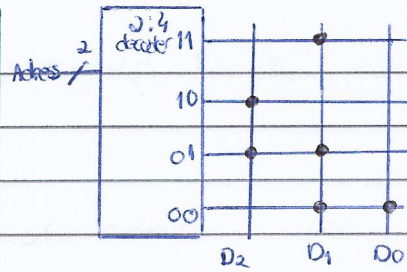
o valor precisa de ser reescrito não só periodicamente  
mas tamb sempre que se lido

estática



bit guardado em dois inversores acoplados  
cada célula precisa de um circuito complexo  
valor mantido enquanto o circuito estiver alimentado

memória  
ROM



Address	Data		
	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

depth

width

$$D_2 = (A_1 \cdot \overline{A_0}) + (\overline{A_1} \cdot A_0) = A_1 \oplus A_0$$

$$D_1 = (\dots) = \overline{A_1} + A_0$$

$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

máquinas de um vários estados possíveis

estado

o estado atual é armazenado num registo (conjunto flip-flops)

estado muda na transição do relógio