

## BLOQUES COMBINACIONALES

### DESCRIPCIÓN:

Son dispositivos asíncronos: no dependen de la señal de sincronización.

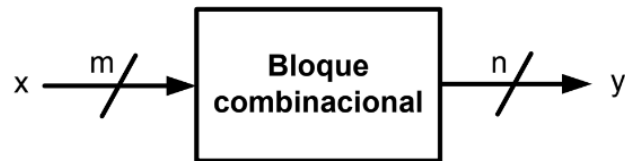
Los bloques combinacionales constan de líneas de entrada y salida.

Se conciben dos tipos de entrada, las entradas de datos y las señales de control.

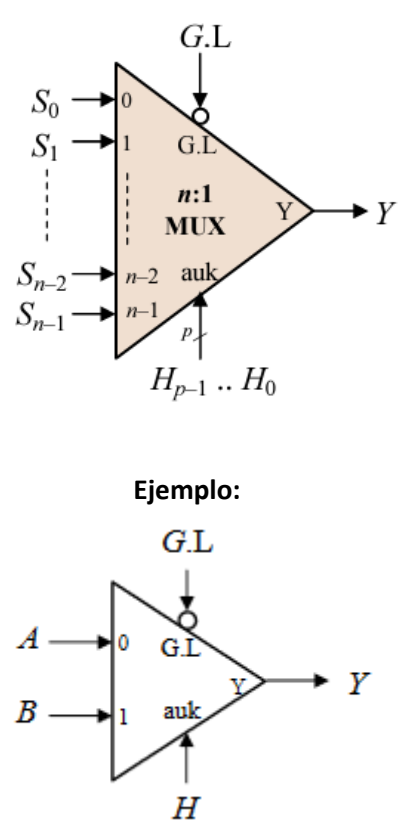
Las salidas en cada momento dependen exclusivamente de los valores de las entradas de ese instante (no tienen memoria, sólo una “combinación”).

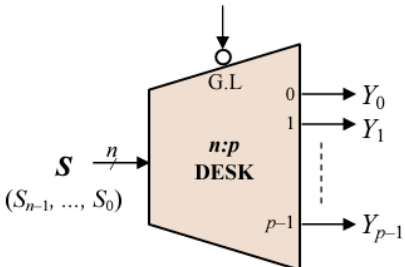
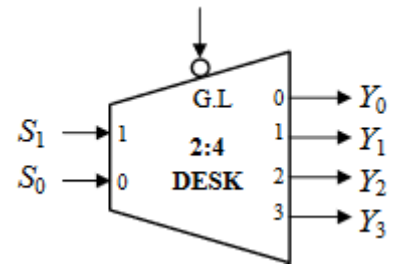
Las señales de control gestionan el funcionamiento del bloque y especifican cómo han de procesarse los datos de entrada para generar los datos de salida pertinentes.

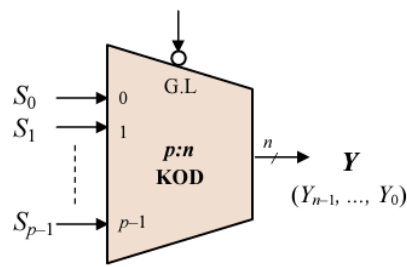
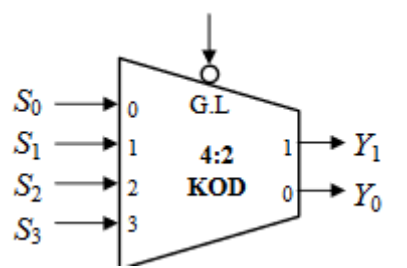
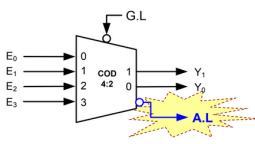
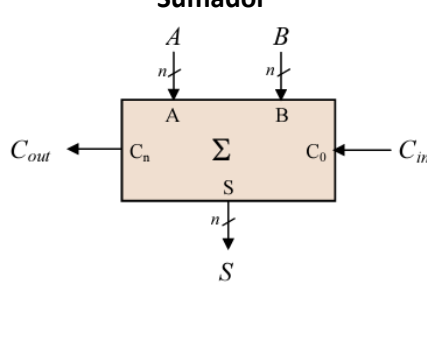
El tiempo invertido en procesar los datos de entrada para producir los datos de salida es conocido como retardo.

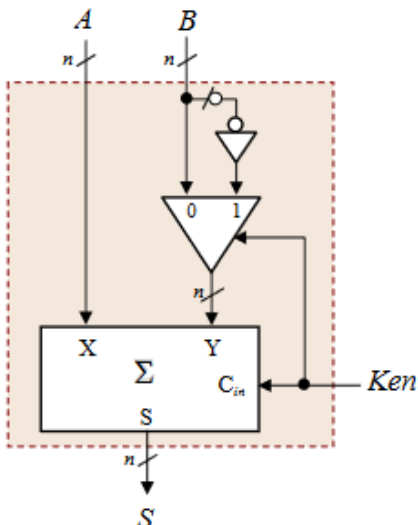


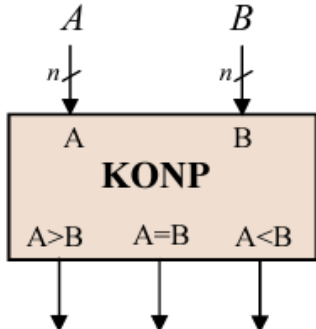
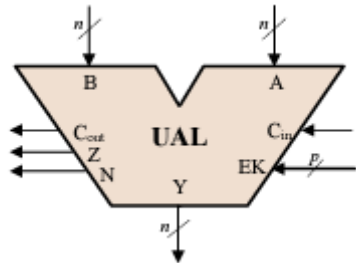
NOMBRE/SÍMBOLO	DESCRIPCIÓN	ENTRADAS	SALIDAS	SEÑALES DE CONTROL
<b>Multiplexor</b> <b>Esquema general:</b>	Bloque combinacional que permite seleccionar cuál de las múltiples entradas transferirá sus datos a la salida del circuito, según lo dictado por las señales de control recibidas.  Tabla de verdad de un multiplexor de dos entradas (A,B):	<b>S:</b> n entradas de 1 bit ( $S_0 S_1 \dots S_{n-1}$ ) siendo $n = 2^p$ .	<b>Y:</b> 1 bit, correspondiente a la entrada seleccionada.	<b>H:</b> señal de selección de p bits. Indica la entrada que se quiere escoger de entre las p posibilidades que nos ofrecen las n entradas, siendo $p = \log_2 n$ .  <b>G:</b> señal de habilitación (enable). - <b>G=0:</b> el multiplexor está desactivado.

 <p><b>Ejemplo:</b></p>	<table border="1" data-bbox="672 239 1030 446"> <thead> <tr> <th><math>G</math></th><th><math>H</math></th><th><math>A</math></th><th><math>B</math></th><th><math>Y</math></th></tr> </thead> <tbody> <tr><td>0</td><td>–</td><td>–</td><td>–</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>–</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>–</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>–</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>–</td><td>1</td><td>1</td></tr> </tbody> </table>	$G$	$H$	$A$	$B$	$Y$	0	–	–	–	0	1	0	0	–	0	1	0	1	–	1	1	1	–	0	0	1	1	–	1	1			<p>- <b>G=1</b>: el multiplexor está activado.</p>
$G$	$H$	$A$	$B$	$Y$																														
0	–	–	–	0																														
1	0	0	–	0																														
1	0	1	–	1																														
1	1	–	0	0																														
1	1	–	1	1																														
<p><b>Descodificador</b> <b>Esquema general:</b></p>	<p>Bloque combinacional cuya función es decodificar información.</p> <p>Lo habitual es pasar dicha información de binario a decimal.</p>	<p><b>S</b>: código binario de n bits a decodificar (<math>S_{n-1}, \dots, S_0</math>)</p>	<p><b>Y</b>: p salidas de un bit</p> <p>Se activa una sola salida de las p posibles, la que corresponde al valor</p>	<p><b>G</b>: señal de habilitación (enable).</p> <p>- <b>G=0</b>: el decodificador está desactivado.</p> <p>- <b>G=1</b>: el decodificador está activado.</p>																														

<div><p><b>Ejemplo:</b></p></div>	<p>Este bloque se usa frecuentemente para...</p> <ul style="list-style-type: none"><li>- decodificar la dirección de una memoria</li><li>- decodificar estados en autómatas (necesario para generar señales de control)</li><li>- decodificar códigos de las instrucciones en los procesadores</li><li>- construir funciones lógicas ...</li></ul> <p>Pueden llevar a cabo funciones de demultiplexores. Son equivalentes pero con distintas señales de entrada, salida y control.</p> <p>Tabla de verdad del de un decodificador 2:4</p> <table><tr><th>G</th><th>S<sub>1</sub></th><th>S<sub>0</sub></th><th>Y<sub>0</sub></th><th>Y<sub>1</sub></th><th>Y<sub>2</sub></th><th>Y<sub>3</sub></th></tr><tr><td>0</td><td>—</td><td>—</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	G	S <sub>1</sub>	S <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	0	—	—	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0	1	1	1	0	0	0	1		<p>decimal del código de entrada.</p> <p><math>p=2^n</math> posibles valores decimales del código binario de entrada.</p>	
G	S <sub>1</sub>	S <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>																																								
0	—	—	0	0	0	0																																								
1	0	0	1	0	0	0																																								
1	0	1	0	1	0	0																																								
1	1	0	0	0	1	0																																								
1	1	1	0	0	0	1																																								
<p><b>Codificador</b> <b>Esquema general</b></p>	<p>Codifican la información de la entrada.</p>	<p><b>S:</b> p entradas de 1 bit (<math>S_0 \dots S_p</math>). Cuando se activa una entrada,</p>	<p><b>Y:</b> código binario de n bits</p>	<p><b>G:</b> señal de habilitación (enable).</p>																																										

<div><p><b>Ejemplo</b></p></div>	<div><p>Lo habitual es pasar dicha información de decimal a binario.</p><p>Cuando se activa una entrada, el codificador devuelve su código binario en la salida.</p><p>Tabla de verdad asociada al ejemplo (4:2):</p><table><tr><th>G</th><th>S<sub>3</sub></th><th>S<sub>2</sub></th><th>S<sub>1</sub></th><th>S<sub>0</sub></th><th>Y<sub>1</sub></th><th>Y<sub>0</sub></th></tr><tr><td>0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table></div>	G	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>	0	—	—	—	—	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0	1	1	<div><p>el codificador devuelve su código binario en la salida.</p><p>Si se activa más de una entrada a la vez se establecen “criterios de prioridad”.</p><p>En general, se codifica la entrada de mayor peso entre las activadas: Codificadores de prioridad.</p></div>	<div><p>correspondiente a la entrada activada.</p><p>Siendo <math>n=\log_2 p</math></p><p><b>A (activada):</b> valdrá 1 si alguna entrada está activada:</p></div>	<div><p>- <b>G=0</b>: el codificador está desactivado.</p><p>- <b>G=1</b>: el codificador está activado.</p></div>
G	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Y <sub>1</sub>	Y <sub>0</sub>																																															
0	—	—	—	—	0	0																																															
1	0	0	0	0	0	0																																															
1	0	0	0	1	0	0																																															
1	0	0	1	0	0	1																																															
1	0	1	0	0	1	0																																															
1	1	0	0	0	1	1																																															
<div><p><b>Sumador</b></p></div>	<div><p>El sumador suma los dos datos de la entrada (A y B) y la llevada (Cin); obteniendo así la suma (S) y la llevada de salida (Cout).</p><p>Tabla de verdad:</p></div>	<div><p><b>A y B:</b> valores (n bits) que se quieren sumar.</p><p><b>Cin:</b> llevada de entrada para la suma (1 bit).</p></div>	<div><p><b>S:</b> suma de los dos valores de la entrada (n bits).</p><p><b>Cout:</b> llevada de salida (1 bit). Indica si se ha producido desbordamiento (overflow).</p></div>	<div><p>No tiene señales de control</p></div>																																																	

	<table><tr><th><math>C_{in}</math></th><th><math>A</math></th><th><math>B</math></th><th><math>C_{out}</math></th><th><math>S</math></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	$C_{in}$	$A$	$B$	$C_{out}$	$S$	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	0	1	0	0	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1	1			
$C_{in}$	$A$	$B$	$C_{out}$	$S$																																													
0	0	0	0	0																																													
0	0	1	0	1																																													
0	1	0	0	1																																													
0	1	1	1	0																																													
1	0	0	0	1																																													
1	0	1	1	0																																													
1	1	0	1	0																																													
1	1	1	1	1																																													
<p><b>Restador</b></p> 	<p>El sumador es válido para hacer sumas y restas (sumador-restador).</p> <p>Para ello se controlan B y Cin de la forma adecuada con la señal de control Ken.</p>	<p><b>A y B:</b> valores a restar (n bits).</p> <p><b>Cin:</b> llevada de entrada (1 bit).</p>	<p><b>S:</b> resta de los dos valores de la entrada (n bits).</p> <p><b>Cout:</b> llevada de salida (1 bit). Indica si se ha producido desbordamiento (overflow).</p>	<p><b>Ken:</b> señal que indica si se ha de realizar la resta (Res=1) o la suma (Res=0).</p> <table><tr><th><math>Ken</math></th><th><math>S</math></th></tr><tr><td>0</td><td><math>A + B</math> (<math>A + B + 0</math>)</td></tr><tr><td>1</td><td><math>A - B</math> (<math>A + \text{not } B + 1</math>)</td></tr></table>	$Ken$	$S$	0	$A + B$ ( $A + B + 0$ )	1	$A - B$ ( $A + \text{not } B + 1$ )																																							
$Ken$	$S$																																																
0	$A + B$ ( $A + B + 0$ )																																																
1	$A - B$ ( $A + \text{not } B + 1$ )																																																
<p><b>Comparador</b></p>	<p>Bloque combinacional que compara valores de entrada.</p> <p>Los comparadores de n bits se construyen con comparadores de un bit (rodajas de bit).</p>	<p><b>A y B:</b> valores a comparar (n bits)</p>	<p>3 salidas de un bit: <b>A&gt;B, A=B, A&lt;B</b></p> <p><b>A&gt;B:</b> indica si A es mayor que B.</p>	<p>No tiene señales de control</p>																																													

	<p>Tabla de verdad:</p> <table border="1" data-bbox="665 271 1046 399"><thead><tr><th><math>A_0</math></th><th><math>B_0</math></th><th><math>(A&gt;B)_{1bit}</math></th><th><math>(A=B)_{1bit}</math></th><th><math>(A&lt;B)_{1bit}</math></th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></tbody></table>	$A_0$	$B_0$	$(A>B)_{1bit}$	$(A=B)_{1bit}$	$(A<B)_{1bit}$	0	0	0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	1	0		<p><b>A=B:</b> indica si A y B son iguales.</p> <p><b>A&lt;B:</b> indica si A es menor que B.</p>	
$A_0$	$B_0$	$(A>B)_{1bit}$	$(A=B)_{1bit}$	$(A<B)_{1bit}$																									
0	0	0	1	0																									
0	1	0	0	1																									
1	0	1	0	0																									
1	1	0	1	0																									
	<p>Bloque para realizar diversas operaciones aritméticas (+, -, +1, -1...) y lógicas (and, or, xor) entre dos valores de entrada.</p> <p>Es el más complejo de los circuitos vistos: LSI.</p> <p>Operaciones habituales:</p> <ul style="list-style-type: none"><li>- Clear (<math>Y = 00 \dots 000</math>).</li><li>- Preset (<math>Y = 11 \dots 111</math>).</li><li>- A, B, A-B, B-A, A+1, A-1, A and B, A xor B, A &gt; B,...</li><li>- Desplazamientos (shift) a la izquierda y a la derecha (multiplicar o dividir por 2),...</li></ul>	<p><b>A y B:</b> valores que se quieren procesar (n bits).</p> <p><b>Cin:</b> llevada de entrada (1 bit).</p>	<p><b>Y:</b> resultado de la operación (n bits)</p> <p>Códigos de condición:</p> <ul style="list-style-type: none"><li>- <b>Cout</b> : llevada de salida de operaciones aritméticas (1 bit).</li><li>- <b>Z</b> : indica si el resultado es cero (1 bit). Si Y=0 entonces Z=1, en caso contrario Z=0.</li><li>- <b>N</b>: Indica si el resultado es negativo (1 bit).</li></ul>	<p><b>EK:</b> código de operación (p bits). Indica qué operación se quiere llevar a cabo a partir de los datos de entrada. Usando p bits se pueden indicar <math>2^p</math> operaciones.</p> <p>Ejemplo de EK:</p> <table border="1" data-bbox="1644 912 1977 1110"><thead><tr><th>Eragiketa-kodea <math>EK_1EK_0</math></th><th>Emaitza (eragiketa)</th></tr></thead><tbody><tr><td>00</td><td><math>A + B</math></td></tr><tr><td>01</td><td><math>A \text{ and } B</math></td></tr><tr><td>10</td><td><math>A \text{ or } B</math></td></tr><tr><td>11</td><td><math>\overline{A}</math></td></tr></tbody></table>	Eragiketa-kodea $EK_1EK_0$	Emaitza (eragiketa)	00	$A + B$	01	$A \text{ and } B$	10	$A \text{ or } B$	11	$\overline{A}$															
Eragiketa-kodea $EK_1EK_0$	Emaitza (eragiketa)																												
00	$A + B$																												
01	$A \text{ and } B$																												
10	$A \text{ or } B$																												
11	$\overline{A}$																												

## BLOQUES SECUENCIALES

### DESCRIPCIÓN:

En los circuitos secuenciales la respuesta depende del valor de las entradas y el estado del sistema.

En circuitos secuenciales se produce **realimentación**: algunas de las salidas del circuito se llevan a la entrada. Gracias a esto obtenemos el “efecto memoria” (capacidad de recordar el valor anterior).

Hay dos tipos de circuitos secuenciales:

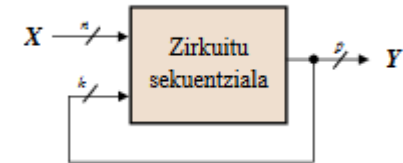
- **Asíncronos**: procesan las señales de entrada en todo momento para ofrecer la salida.
- **Síncronos**: procesan las entradas en los intervalos/instantes de tiempo señalados por una señal de control: reloj (clock, clk). Fuera de dichos intervalos/instantes la salida del circuito no variará, aunque cambien las entradas.




Los circuitos producen retardos. Pueden generar salidas erróneas (glitches) durante un tiempo determinado. Si las señales no se procesan en cualquier momento (síncr.), se minimiza la influencia de posibles “ruidos” o “glitches”.

El **reloj** es la señal de control principal en los circuitos secuenciales. Define la “velocidad” del circuito y oscila continuamente entre los valores 0 y 1. En general, se usa el flanco de subida para procesar las entradas.

Características (magnitudes):

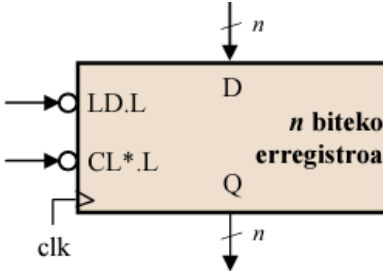
- **periodo (T)**: se mide en segundos.
- **frecuencia (f)**: se mide en Hercios (HZ= ciclos/s = s<sup>-1</sup>)

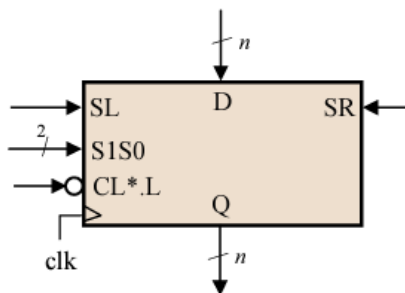


NOMBRE	DESCRIPCIÓN	ENTRADAS	SALIDAS	SEÑALES DE CONTROL																	
<b>Biestable JK</b>	<p>Biestable síncrono con un bit de memoria. Puede estar en dos estados: 1 o 0.</p> <p>Es usado para construir indicadores (activar o desactivar información de control de 1 bit).</p> <p>Tabla de verdad:</p>	<p>Dos entradas síncronas:</p> <p>- <b>J</b>: para activar señales de 1 bit, se ofrece en H.</p> <p>- <b>K</b>: para desactivar señales de 1 bit, se ofrece en L.</p>	<p><b>Q</b>: salidas de datos (1 bit), se suele ofrecer en H y L.</p> <table><tr><th>clk</th><th>J</th><th>K</th><th>Q<sub>n+1</sub></th></tr><tr><td rowspan="4"></td><td>0</td><td>0</td><td>Q<sub>n</sub></td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td><math>\overline{Q_n}</math></td></tr></table>	clk	J	K	Q <sub>n+1</sub>		0	0	Q <sub>n</sub>	0	1	0	1	0	1	1	1	$\overline{Q_n}$	<p>2 señales de control asíncronas:</p> <p><b>CL*</b>: Clear o puesta a 0 (Reset).</p> <p><b>PR*</b>: Preset o puesta a 1 (Set).</p> <p><b>Notas:</b></p> <p>- al ser señales asíncronas su efecto es inmediato e independiente a clk.</p> <p>- no han de activarse ambas señales asíncronas a la vez, el estado quedaría indeterminado.</p>
clk	J	K	Q <sub>n+1</sub>																		
	0	0	Q <sub>n</sub>																		
	0	1	0																		
	1	0	1																		
	1	1	$\overline{Q_n}$																		

	<table><tr><th>CL*</th><th>PR*</th><th>clk</th><th>J</th><th>K</th><th>Q<sub>t+1</sub></th></tr><tr><td>1</td><td>0</td><td>—</td><td>—</td><td>—</td><td>0</td></tr><tr><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td><td>1</td></tr><tr><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">↗</td><td>0</td><td>0</td><td>Q<sub>t</sub></td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>Q̄<sub>t</sub></td></tr></table>	CL*	PR*	clk	J	K	Q <sub>t+1</sub>	1	0	—	—	—	0	0	1	—	—	—	1	0	0	↗	0	0	Q <sub>t</sub>	0	1	0	1	0	1	1	1	Q̄ <sub>t</sub>			
CL*	PR*	clk	J	K	Q <sub>t+1</sub>																																
1	0	—	—	—	0																																
0	1	—	—	—	1																																
0	0	↗	0	0	Q <sub>t</sub>																																
			0	1	0																																
			1	0	1																																
			1	1	Q̄ <sub>t</sub>																																
<p><b>Biestable D</b></p>	<p>Biestable síncrono con un bit de memoria. Puede estar en dos estados: 1 o 0.</p> <p>Posiblemente el biestable más usado.</p> <p>Es usado para...</p> <ul style="list-style-type: none"><li>- Guardar información.</li><li>- Sincronizar señales externas.</li><li>- guardar los estados de las unidades de control de sistemas digitales.</li></ul> <p>Tabla de verdad:</p>	<p><b>D</b>: entrada síncrona de datos (1 bit).</p>	<p><b>Q</b>: salida de datos (1 bit), se suele ofrecer en H y L.</p> <table><tr><th>clk</th><th>D</th><th>Q<sub>t+1</sub></th></tr><tr><td rowspan="2">↗</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	clk	D	Q <sub>t+1</sub>	↗	0	0	1	1	<p>2 Señales de control asíncronas:</p> <p><b>CL*</b>: Clear o puesta a 0 (Reset).</p> <p><b>PR*</b>: Preset o puesta a 1 (Set).</p> <p><b>Notas:</b></p> <ul style="list-style-type: none"><li>- al ser señales asíncronas su efecto es inmediato e independiente a clk.</li><li>- no han de activarse ambas señales asíncronas a la vez, el estado quedaría indeterminado.</li></ul>																									
clk	D	Q <sub>t+1</sub>																																			
↗	0	0																																			
	1	1																																			



	<table><tr><th>CL*</th><th>PR*</th><th>clk</th><th>D</th><th>Q<sub>t+1</sub></th></tr><tr><td>1</td><td>0</td><td>–</td><td>–</td><td>0</td></tr><tr><td>0</td><td>1</td><td>–</td><td>–</td><td>1</td></tr><tr><td>0</td><td>0</td><td>⌋</td><td>0</td><td>0</td></tr><tr><td></td><td></td><td></td><td>1</td><td>1</td></tr></table>	CL*	PR*	clk	D	Q <sub>t+1</sub>	1	0	–	–	0	0	1	–	–	1	0	0	⌋	0	0				1	1			
CL*	PR*	clk	D	Q <sub>t+1</sub>																									
1	0	–	–	0																									
0	1	–	–	1																									
0	0	⌋	0	0																									
			1	1																									
<div><p><b>Registro</b></p></div>	<p>Dispositivo síncrono capaz de almacenar un dato de n bits indefinidamente.</p> <p>Generalmente se construye con biestables D habilitados.</p> <p>Tabla de verdad:</p> <table><tr><th>CL*</th><th>clk</th><th>LD</th><th>Q<sub>t+1</sub></th></tr><tr><td>1</td><td>–</td><td>–</td><td>0</td></tr><tr><td>0</td><td>⌋</td><td>0</td><td>Q<sub>t</sub></td></tr><tr><td>0</td><td></td><td>1</td><td>D</td></tr></table>	CL*	clk	LD	Q <sub>t+1</sub>	1	–	–	0	0	⌋	0	Q <sub>t</sub>	0		1	D	<p><b>D</b>: entrada de datos donde se ha de colocar el dato a cargar en el registro (n bits).</p>	<p><b>Q</b>: salida de datos en la que disponemos siempre del contenido del registro (n bits).</p>	<p><b>LD</b>: señal síncrona que indica que se debe cargar un dato en el registro (load: cargar). Al llegar el flanco de reloj:</p> <ul style="list-style-type: none"><li>- <b>LD=1</b>: se carga un nuevo dato,</li><li>- <b>LD=0</b>: se mantiene el contenido.</li></ul> <p><b>CL*</b>: señal de inicialización (clear).</p> <ul style="list-style-type: none"><li>- Permite poner a 0 el contenido del registro.</li><li>- Normalmente es una señal asíncrona.</li></ul>									
CL*	clk	LD	Q <sub>t+1</sub>																										
1	–	–	0																										
0	⌋	0	Q <sub>t</sub>																										
0		1	D																										
<p><b>Registros de desplazamiento</b></p>	<p>Comportamiento:</p> <ul style="list-style-type: none"><li>- Almacenar un dato de n bits.</li><li>- Desplazar el contenido almacenado 1 bit a la izquierda (multiplicar por 2) o a la derecha (dividir por 2).</li></ul>	<p><b>D</b>: entrada de datos donde se ha de colocar el dato a cargar en el registro (n bits).</p>	<p><b>Q</b>: salida de datos en la que disponemos siempre del contenido del registro (n bits).</p>	<p><b>S1, S0</b>: dos bits para indicar la operación a realizar (cargar un dato, mantenerlo o desplazarlo a la derecha o izquierda)</p> <p><b>CL*</b>: señal de inicialización (clear).</p>																									



Operaciones básicas con Registros de Desplazamiento:

- Multiplicar / Dividir por 2
- Rotación de bits

Generalmente se construye con biestables D habilitados.

Se usan en...

- sistemas digitales donde haya que procesar información en serie (bit a bit).
- circuitos digitales donde se gestionen divisiones y multiplicaciones

Tabla de verdad:

clk	S1	S0	$Q (Q_{n-1} \dots Q_0)$
⌋	0	0	$Q_{n-1} \dots Q_0$
	0	1	$Q_{n-2} \dots Q_0$ <b>SR</b>
	1	0	<b>SL</b> $Q_{n-1} \dots Q_1$
	1	1	$D_{n-1} \dots D_0$

**SL:** entrada de 1 bit (left=izda), se cargará en el bit de más peso si el dato se desplaza hacia la derecha

**SR:** entrada de 1 bit (right=dcha), se cargará en el bit de menos peso si el dato se desplaza hacia la izquierda

- Permite poner a 0 el contenido del registro.
- Normalmente es una señal asíncrona.

<div>Contadores</div> <div></div>	<div>Comportamiento:</div> <div><ul style="list-style-type: none"><li>- Almacenar un dato de n bits</li><li>- Ejecutar operaciones aritméticas sencillas con el contenido: +1 y -1 (contar).</li></ul></div> <div>En general, los contadores son bidireccionales: Up/Down. Algunos no tienen señal UP y no se puede elegir el sentido (cuentan en una sola dirección). Pueden darse contadores sin señal CL.</div> <div>Un contador físico es finito, existe un valor máximo que puede representar.</div> <div>Contador binario de n bits: contador módulo 2n {0,1,2,...,2n-1}.</div> <div>Tabla de verdad:</div> <table><tr><th>CL*</th><th>clk</th><th>LD</th><th>E</th><th>UP</th><th>Q<sub>t+1</sub></th></tr><tr><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td></tr><tr><td>0</td><td rowspan="4"></td><td>0</td><td>0</td><td>-</td><td>Q<sub>t</sub></td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>(Q<sub>t</sub>-1) mod 2<sup>n</sup></td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>(Q<sub>t</sub>+1) mod 2<sup>n</sup></td></tr><tr><td>0</td><td>1</td><td>-</td><td>-</td><td>D</td></tr></table>	CL*	clk	LD	E	UP	Q <sub>t+1</sub>	1	-	-	-	-	0	0		0	0	-	Q <sub>t</sub>	0	0	1	0	(Q <sub>t</sub> -1) mod 2 <sup>n</sup>	0	0	1	1	(Q <sub>t</sub> +1) mod 2 <sup>n</sup>	0	1	-	-	D	<div>D: entrada de datos donde se ha de colocar el dato a cargar en el registro (n bits).</div> <div>TC: (1 bit) indica si la cuenta ha llegado al límite (Terminal Count). También se le llama RCO (Ripple Carry Output).</div> <div><ul style="list-style-type: none"><li>- Si cuenta <b>ascendente: TC=1</b> si todos los bits salida son "1".</li><li>- Si cuenta <b>descendente: TC=1</b> si todos los bits salida son "0".</li></ul></div>	<div>LD: señal de control de 1 bit, indica que se debe cargar un dato.</div> <div>E (CET): señal de control de 1 bit, indica que se quiere contar. En caso de encadenamiento de contadores se usa para trasladar el TC.</div> <div>UP (U/D, U/D.L): indica la dirección de la cuenta:</div> <div><ul style="list-style-type: none"><li>- UP=1: hacia arriba.</li><li>- UP=0: hacia abajo.</li></ul></div> <div>CL*: Señal de inicialización (clear).</div> <div><ul style="list-style-type: none"><li>- Permite poner a 0 el contenido.</li><li>- Normalmente es una señal asíncrona.</li></ul></div>
CL*	clk	LD	E	UP	Q <sub>t+1</sub>																															
1	-	-	-	-	0																															
0		0	0	-	Q <sub>t</sub>																															
0		0	1	0	(Q <sub>t</sub> -1) mod 2 <sup>n</sup>																															
0		0	1	1	(Q <sub>t</sub> +1) mod 2 <sup>n</sup>																															
0		1	-	-	D																															

## MEMORIAS

### DESCRIPCIÓN:

Las memorias nacen de la necesidad de almacenar una cantidad de información mayor.

La información que se guarda en una memoria se estructura en **palabras** (datos de n bits, generalmente 8).

A cada palabra o posición le corresponde una **dirección** (utilizada para operaciones sobre memoria: rd/wr). Se dan en binario y ocupan p bits.

Siendo la cantidad de palabras almacenable: N, se cumple  $p = \log_2 N$  o  $N = 2^p$ .

Principales parámetros:

- **Capacidad (bits almacenables):** cantidad de palabras  $\times$  tamaño de las palabras
- **Tiempo de respuesta/acceso:** tiempo para una operación rd/wr.
- **Tiempo de ciclo de lectura o escritura:** tiempo mínimo entre dos operaciones consecutivas.

Cuanto mayor es una memoria, mayor es su tiempo de respuesta.

Cuando el tiempo de respuesta es importante se construye una jerarquía de memorias:

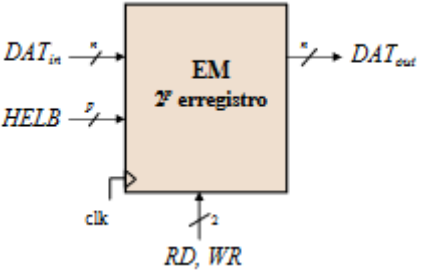
- Los datos más usados se guardan en memorias más pequeñas y rápidas.
- La información general se guarda en memorias más grandes y lentas.

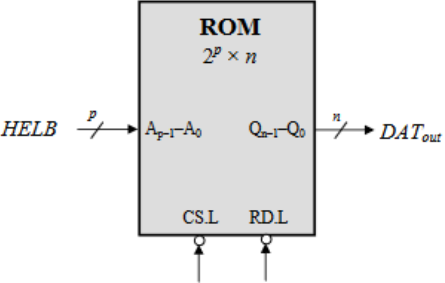
Sistema de memoria formado por 3 tipos de memorias semiconductoras:

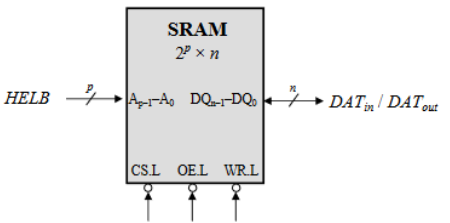
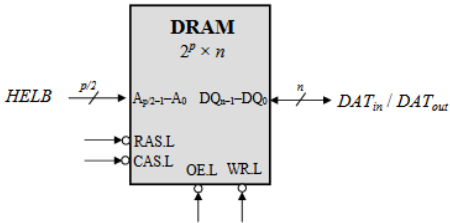
- **Registros:** más rápidos, baja capacidad.
- **Memoria caché:** capacidad y velocidad intermedias
- **Memoria principal:** la más lenta, mayor capacidad.

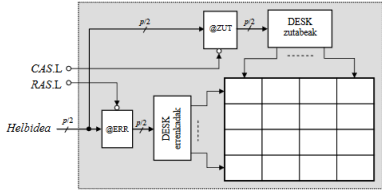
Características ideales:

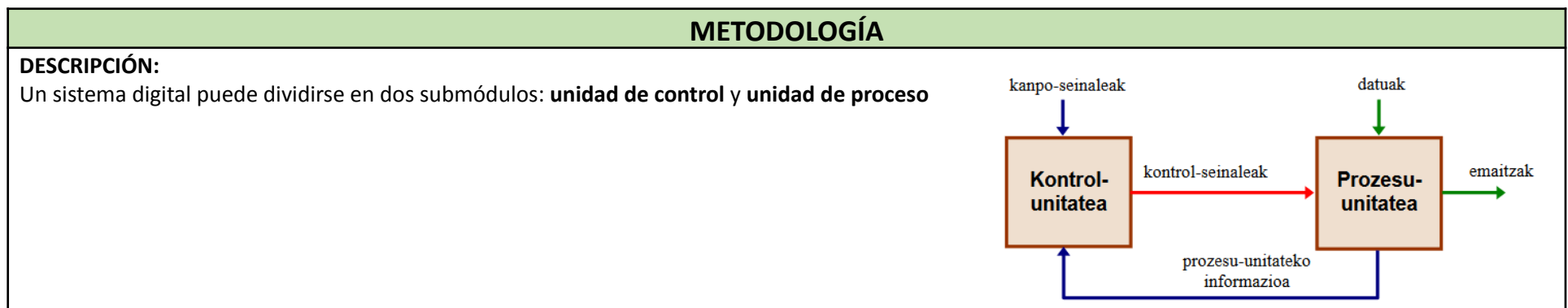
- Rapidez.
- Permite operaciones rd y wr.
- Información permanente (se mantiene al desconectarse la fuente de alimentación).
- Alta densidad (muchos bits en un solo circuito integrado).
- Fiable (sin errores en la información almacenada).

NOMBRE	DESCRIPCIÓN	ENTRADAS	SALIDAS	SEÑALES DE CONTROL												
<div><b>BANCO DE REGISTROS</b> </div>	<p>Pequeña memoria formada por unos cuantos registros donde se guarda/recupera información.</p> <p><b>Muy rápido</b> pero de <b>baja capacidad</b> (4-256 datos).</p> <p>Permite operaciones <b>rd/wr</b>.</p> <p>A cada registro se le asigna una dirección y un “nombre” dentro de la arquitectura (ej: dir 7, R7).</p> <p><b>Proceso de escritura:</b></p> <ul style="list-style-type: none"><li>- dar el dato (<b>DAT<sub>in</sub></b>)</li><li>- indicar la dirección donde queremos escribir (<b>HELB</b>)</li><li>- activar la señal <b>WR</b>.</li></ul> <p><b>Proceso de lectura:</b></p> <ul style="list-style-type: none"><li>- indicar la dirección desde donde queremos leer (<b>HELB</b>)</li><li>- activar la señal <b>RD</b>.</li><li>- se nos proporciona el dato <b>DAT<sub>out</sub></b>.</li></ul> <p>Utilizado por el procesador para ejecutar instrucciones más rápidamente.</p>	<p><b>DAT<sub>in</sub></b>: dato a escribir en un registro (n bits).</p> <p><b>HELB</b>: dirección del registro a leer/escribir (n bits).</p>	<p><b>DAT<sub>out</sub></b>: dato a leer del registro especificado.</p>	<p><b>RD</b>: señal que indica que se va a realizar una lectura.</p> <p><b>WR</b>: señal que indica que se va a realizar una escritura.</p> <table><tr><th>RD</th><th>WR</th><th>Eragiketa</th></tr><tr><td>0</td><td>0</td><td>informazioa mantendu</td></tr><tr><td>0</td><td>1</td><td>erregistro bat idatzi EM[HELB] := DAT<sub>in</sub></td></tr><tr><td>1</td><td>0</td><td>erregistro bat irakurri DAT<sub>out</sub> := EM[HELB]</td></tr></table>	RD	WR	Eragiketa	0	0	informazioa mantendu	0	1	erregistro bat idatzi EM[HELB] := DAT <sub>in</sub>	1	0	erregistro bat irakurri DAT <sub>out</sub> := EM[HELB]
RD	WR	Eragiketa														
0	0	informazioa mantendu														
0	1	erregistro bat idatzi EM[HELB] := DAT <sub>in</sub>														
1	0	erregistro bat irakurri DAT <sub>out</sub> := EM[HELB]														

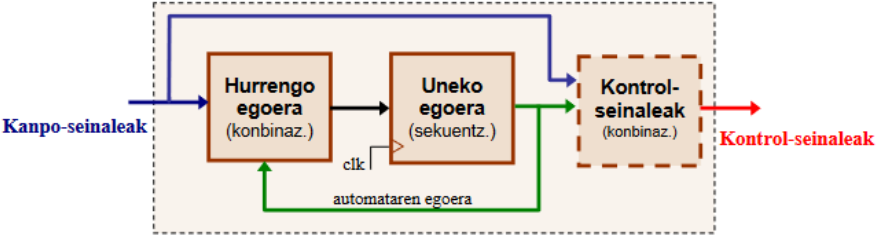
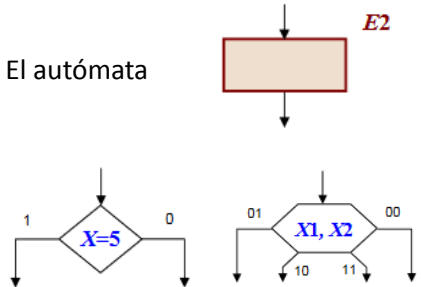
<p><b>MEMORIA ROM (read only memory)</b></p> 	<p>Permite operación <b>rd</b> generalmente.</p> <p>En algunas se puede escribir pero lleva más tiempo.</p> <p>La información que almacenan es <b>permanente</b> y en general se graba fuera del sistema en el que se utilizan.</p> <p>La información que guardan es fija o cambia sólo esporádicamente.</p>	<p><b>HEL B</b>: dirección del registro a leer (n bits).</p>	<p><b>DAT<sub>out</sub></b>: dato a leer de la memoria.</p>	<p><b>RD (read)</b>: señal a activar para realizar una lectura (en algunos casos: <b>OE</b>).</p> <p><b>CS (chip select)</b>: señal de activación del chip.</p> <ul style="list-style-type: none"> <li>- <b>CS=1</b>: memoria activada</li> <li>- <b>CS=0</b>: memoria desactivada</li> </ul>
<p><b>MEMORIA RAM (Random Access Memory)</b></p>	<p>Permite operaciones <b>rd/wr</b>.</p> <p>La información que almacenan <b>no</b> es <b>permanente</b> (no se mantiene al desconectarse la fuente de alimentación).</p> <p>El tiempo para acceder a cualquier palabra es el mismo, independientemente de su posición.</p> <p>Hay dos tipos: RAM estática y dinámica.</p>			

<p><b>MEMORIA RAM ESTÁTICA</b></p>  <p>The diagram shows a block labeled 'SRAM' with capacity <math>2^p \times n</math>. It has a <math>p</math>-bit address input <math>HELB</math> connected to lines <math>A_{p-1}-A_0</math>. It has <math>n</math>-bit data lines <math>DQ_{n-1}-DQ_0</math> connected to <math>DAT_{in} / DAT_{out}</math>. Control lines at the bottom are <math>CSL</math>, <math>OEL</math>, and <math>WRL</math>, each with a bubble indicating active-low signals.</p>	<p><b>Lectura:</b> se ha de proporcionar la dirección antes que la señal OE, o bien al mismo tiempo.</p> <p><b>Escritura:</b> se ha de proporcionar el dato (<b>DAT<sub>in</sub></b>) y la dirección (<b>HELB</b>) y activar WR.</p>	<p>Entradas y salidas:</p> <p><b>HELB:</b> dirección de <math>p</math> bits de una palabra concreta sobre la que se realizará la operación (entrada). La memoria tendrá <math>2^p</math> palabras.</p> <p><b>DAT<sub>out</sub> / DAT<sub>in</sub>:</b> dato de <math>n</math> bits a leer/escribir respectivamente. Se utilizan las mismas líneas para los datos de entrada y salida. Cuando <b>CS=0</b> o no se realizan operaciones las líneas de datos están en alta impedancia (<b>Z</b>).</p>	<p><b>OE (Output Enable):</b> señal a activar para realizar una lectura. Hace que las líneas de datos funcionen como salida.</p> <p><b>WR (write):</b> señal a activar para realizar una escritura. Hace que las líneas de datos funcionen como entrada.</p> <p><b>CS (chip select):</b> señal de activación del chip.</p> <ul style="list-style-type: none"> <li>- <b>CS=1:</b> memoria activada</li> <li>- <b>CS=0:</b> memoria desactivada</li> </ul> <table border="1" data-bbox="1653 821 2060 981"> <thead> <tr> <th>CS</th><th>WR</th><th>OE</th><th>funtzioa</th></tr> </thead> <tbody> <tr> <td>0</td><td>—</td><td>—</td><td>desaktibatuta</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>—</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>irakurketa</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>idazketa</td></tr> </tbody> </table>	CS	WR	OE	funtzioa	0	—	—	desaktibatuta	1	0	0	—	1	0	1	irakurketa	1	1	0	idazketa
CS	WR	OE	funtzioa																				
0	—	—	desaktibatuta																				
1	0	0	—																				
1	0	1	irakurketa																				
1	1	0	idazketa																				
<p><b>MEMORIA RAM DINÁMICA</b></p>  <p>The diagram shows a block labeled 'DRAM' with capacity <math>2^p \times n</math>. It has a <math>p/2</math>-bit address input <math>HELB</math> connected to lines <math>A_{p/2-1}-A_0</math>. It has <math>n</math>-bit data lines <math>DQ_{n-1}-DQ_0</math> connected to <math>DAT_{in} / DAT_{out}</math>. Control lines at the bottom are <math>RASL</math>, <math>CASL</math>, <math>OEL</math>, and <math>WRL</math>, each with a bubble indicating active-low signals.</p>	<p><b>Mayor capacidad</b> que en la estática pero se ha de recargar para evitar pérdidas (<b>refresh</b>).</p> <p>Organizada como una matriz:</p>	<p>Entradas y salidas</p> <p><b>HELB:</b> dirección de <math>p</math> bits de una palabra concreta sobre la que se realizará la operación (entrada). La memoria tendrá <math>2^p</math> palabras.</p> <p><b>DAT<sub>out</sub> / DAT<sub>in</sub>:</b> dato de <math>n</math> bits a leer/escribir respectivamente. Se utilizan las mismas líneas para los datos de entrada y salida.</p>	<p><b>OE (Output Enable):</b> señal a activar para realizar una lectura. Hace que las líneas de datos funcionen como salida.</p> <p><b>WR (write):</b> señal a activar para realizar una escritura. Hace que las líneas de datos funcionen como entrada.</p>																				

	 <p>Procedimiento:</p> <ul style="list-style-type: none"> <li>- La dirección se proporciona en dos fases: primero la dirección de fila (con la señal <b>RAS</b>), después de columna (con la señal <b>CAS</b>).</li> <li>- indicar operación (y <b>DATin</b> de ser WR=1).</li> <li>- desactivar <b>CAS</b> y <b>RAS</b></li> </ul>	<p>Cuando <b>CS=0</b> o no se realizan operaciones las líneas de datos están en alta impedancia (<b>Z</b>).</p>	<p><b>CS</b> (chip select): señal de activación de chip.</p> <ul style="list-style-type: none"> <li>- CS=1 memoria activada</li> <li>- CS=0 memoria desactivada</li> </ul> <p><b>RAS (row address strobe):</b> indica que la dirección corresponde a una fila.</p> <p><b>CAS (column address strobe):</b> indica que la dirección corresponde a una columna.</p>
--	---	---	--





NOMBRE	DESCRIPCIÓN
<b>UP (unidad de proceso)</b>	<p>Trata los datos.</p> <p>El funcionamiento de sus componentes es controlado por la UC.</p> <p>Cada sistema digital está enfocado a tareas diferentes, por lo que variarán de un diseño a otro.</p> <p>No es posible automatizar un diseño que sirva para cualquier sistema digital.</p>
<b>UC (unidad de control)</b>	<p>Tareas similares en cada sistema digital:</p> <p>-&gt; generar las señales de control de los componentes de la UP en una secuencia precisa para lograr su objetivo.</p> <p>En general es un autómata o máquina de estados finitos: realiza las tareas previstas en una secuencia determinada de modo cíclico y continuo.</p> <p>En cada instante de tiempo está en un estado determinado.</p> <p>En cada estado:</p> <ul style="list-style-type: none"> <li>- genera señales de control correspondientes a dicho estado.</li> <li>- decide cuál será el estado siguiente.</li> </ul> <p>Maneras de describir las tareas:</p> <ul style="list-style-type: none"> <li>- grafos ASM (Algorithmic State Machines)</li> <li>- métodos basados en lenguajes de descripción de hardware (VHDL...)</li> </ul> 
<b>ASM (algorithmic State Machines)</b>	<p>Utilizados en sistemas digitales síncronos para describir las tareas del autómata de control.</p> <p>Describe un algoritmo o procedimiento de control.</p> <p>Elementos:</p> <ul style="list-style-type: none"> <li>- <b>Estados:</b> elementos básicos, representados con un rectángulo. Corresponden a los estados del autómata. El autómata está siempre en algún estado y permanece en él al menos un ciclo. Es conveniente asignarles nombres.</li> <li>- <b>Entradas (variables de entrada):</b> procesadas para decidir el siguiente estado y generar las correspondientes señales de control. Representadas por rombos o similares. En cada estado se pueden procesar 0, 1 o varias.</li> </ul> 

- **Señales de control (salidas):**

- **Incondicionales:** se activan siempre que el autómata esté en el estado en el que están indicadas

- **Condicionales:** se activan cuando el autómata está en el estado en el que están indicadas y las entradas toman un valor determinado.

La señal de inicialización inicializa la unidad de control en un estado concreto (generalmente: **Reset**).

Habitualmente, esta señal es asíncrona (en lógica negativa) y se suele llevar a la entrada CL\* de los biestables.

Cuando se llega al flanco del reloj, el algoritmo procesa las variables de entrada.

Determina las señales de control a activar en cada instante y la evolución del propio autómata.

Si las señales no están expresadas en un estado determinado, estarán desactivadas.

Los nombres de las señales de control han de corresponderse con las entradas de control de los circuitos de la unidad de proceso.

