

TRABAJO PRÁCTICO N°1

Decoding and Timing

Grupo II

AUTORES:

Pablo Martín SCHEINFELD (59065),
Santiago Agustín ARRIBERE (59169),
Matías Santiago FRANCOIS (59828),
Carlos Javier MASELLI (59564),
Malena MÜLLER (57057)

PROFESORES:

Daniel Andrés JACOBY,
Nicolas MAGLIOLA,
Diego Matías ISMIRLIAN

Contenido

1. Ejercicio 1	2
1.1. Implementación mediante lógica discreta	2
1.2. Implementación mediante lógica de baja complejidad	4
1.3. Implementación mediante una PAL	6
1.4. Diagrama de tiempos para lectura y escritura	7
2. Ejercicio 2	8
3. Ejercicio 3	11
4. Ejercicio 4	13
5. Ejercicio 5	17

1. Ejercicio 1

El presente ejercicio consiste en diseñar el decodificador de direcciones requerido para poder conectar una memoria RAM de 8K al bus del HC11 a partir de la posición \$4000.

Con el fin de diseñar el decodificador de direcciones se asume que el bus de direcciones es de 16 bits. De esta forma, dicho bus se podría dividir en 8 partes iguales, cada una para un periférico determinado. Así, se pueden determinar 8 salidas del decodificador: desde O7 hasta O0. La salida O0 corresponde a las direcciones entre 0 y \$1FFF; la O1 a las que se encuentran entre \$2000 y \$3FFF; la O2 -la cual cubre el rango previsto para la memoria RAM- desde la posición \$4000 a \$5FFF y así sucesivamente. Mediante esta configuración la señal O2 debe ser conectada en el CS de la memoria RAM para lograr el objetivo buscado.

De esta forma, se tomarán los 3 bits más significativos del bus de direcciones como entrada al decodificador, dado que los otros 13 se utilizan para encontrar la dirección precisa dentro de cada periférico. El decodificador resultante consta entonces de 3 entradas: In15, In14 y In13LSB; y 8 salidas: desde O7 hasta O0. Cabe aclarar que al indicar "In13LSB" se hace referencia a que este es el bit menos significativo entre las 3 entradas del decodificador pero no lo es con respecto a todo el bus de direcciones. Por otra parte, las salidas del decoder -considerándolas activo bajo- deben poder ser habilitadas siempre y cuando el flag E -considerándolo activo alto- se encuentre activo, dado que este indica el lapso de tiempo en el cual se pueden leer o escribir datos en los periféricos. En caso contrario, todas las salidas deben estar desactivadas (valor 1).

1.1. Implementación mediante lógica discreta

A fin de implementar el decodificador mediante lógica discreta se desarrolla una tabla de verdad la cual relaciona los estados de los bits de entrada con los de salida. Esta se expone en la tabla 1.

E	In15	In14	In13LSB	O7	O6	O5	O4	O3	O2	O1	O0
0	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1

TABLA 1: Tabla de verdad del circuito requerido.

A partir de esta tabla de verdad se podrían desarrollar los mapas de Karnaugh resultantes para cada bit de salida. Sin embargo, como cada bit de salida toma el valor 0 (activo) ante solo una de las combinaciones de la entrada y el valor 1 en cualquier caso contrario, realizar los mapas sería redundante. Esto es así dado que si se analiza el problema de acuerdo a productos de sumas, en cada mapa de Karnaugh surge un solo max-término. A modo de ejemplo se muestra en la figura 1 el mapa resultante para la salida O5 -aunque es análogo para todos los bits- y su correspondiente expresión.

		In14, In13LSB			
		00	01	11	10
E, In15	00	1	1	1	1
	01	1	1	1	1
	11	1	0	1	1
	10	1	1	1	1

$\bar{E} + \bar{In15} + In14 + \bar{In13LSB}$

FIGURA 1: Mapa de Karnaugh por producto de sumas de la salida O5 y su expresión lógica.

De esta forma, en la tabla 2 se exponen las expresiones para cada bit de salida.

Bit	Expresión
O0	$\bar{E} + In15 + In14 + In13LSB$
O1	$\bar{E} + In15 + In14 + \bar{In13LSB}$
O2	$\bar{E} + In15 + \bar{In14} + In13LSB$
O3	$\bar{E} + In15 + \bar{In14} + \bar{In13LSB}$
O4	$\bar{E} + \bar{In15} + In14 + In13LSB$
O5	$\bar{E} + \bar{In15} + In14 + \bar{In13LSB}$
O6	$\bar{E} + \bar{In15} + \bar{In14} + In13LSB$
O7	$\bar{E} + \bar{In15} + \bar{In14} + \bar{In13LSB}$

TABLA 2: Expresiones lógicas de los bits de salida del decodificador.

A partir de las expresiones anteriores se puede diagramar el circuito lógico resultante (fig 2).

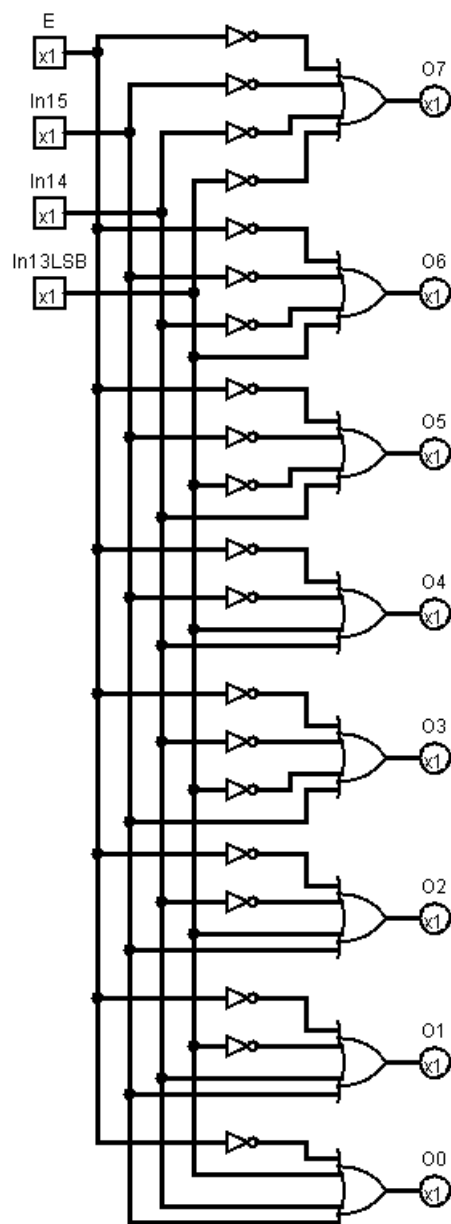


FIGURA 2: Circuito lógico del decodificador.

1.2. Implementación mediante lógica de baja complejidad

En este apartado se analiza como implementar el decodificador requerido mediante lógica de baja complejidad. Para ello se hace uso del integrado 74LS138. En la figura 3 se muestra el pinout de este componente. Como se puede apreciar, el componente es un decodificador de 3x8, cuyas salidas son activo bajo. A su vez, cuenta con tres pines, dos de ellos activo bajo, para controlar el enable. De esta forma, el componente decodifica las entradas cuando el enable (una compuerta AND entre los 3 pines mencionados) toma el valor 1.

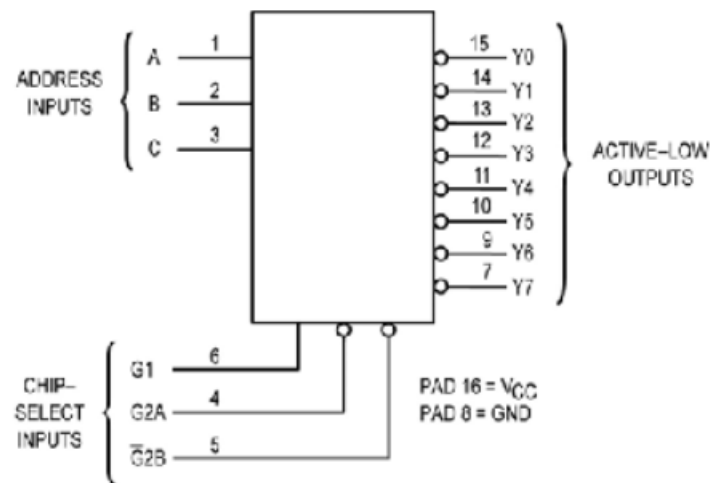


FIGURA 3: Pinout del decoder 74LS138.

Considerando el diagrama anterior, se procedió a determinar a que señal de las mencionadas anteriormente corresponde cada pin del 74LS138. Para ello se analizó la tabla de verdad provista por el fabricante en la hoja de datos, la cual se muestra en la figura 4.

INPUTS						OUTPUTS							
G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

FIGURA 4: Tabla de verdad del decoder 74LS138.

Contrastando dicha tabla con la tabla 1, se concluye que cada salida del integrado "Yx" corresponde a la señal "Ox". Por su parte, los pines A, B y C corresponden respectivamente a las señales In13LSB, In14 e In15. Bajo la finalidad de que el componente funcione en los momentos adecuados, los pines G2A y $\overline{G2B}$ se deben conectar a tierra. Por su parte, la señal E debe introducirse en el pin G1 para que las tablas de verdad sean análogas. A partir de dicho análisis se elaboró el diagrama de la figura 5, donde se indica a que señal corresponde cada bit del integrado.

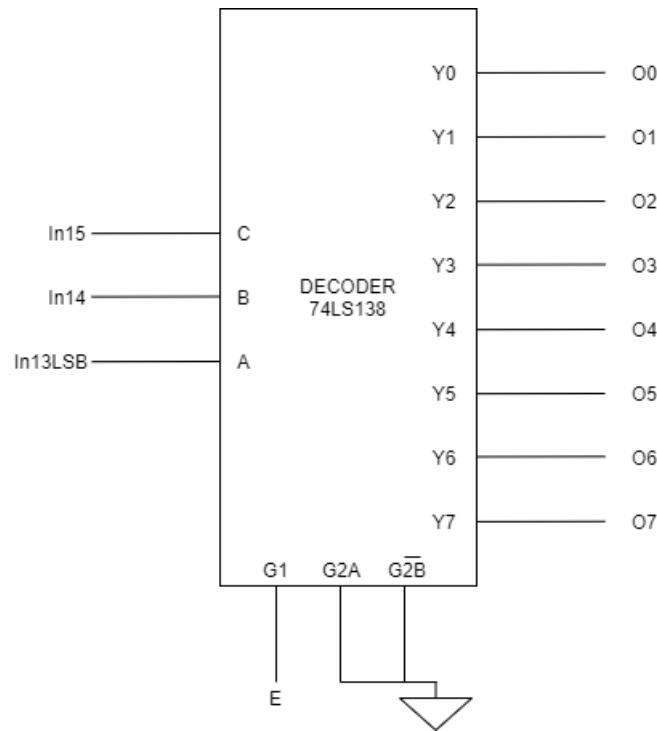


FIGURA 5: Conexionado de los pines del 74LS138 con las señales utilizadas.

Como fue mencionado, la señal O2 puede ser conectada al CS de la memoria RAM de 8K para conectarla a partir de la posición \$4000. Así, queda definido el modo de implementar el decodificador mediante lógica de baja complejidad utilizando el integrado 74LS138.

1.3. Implementación mediante una PAL

Para lograr implementar el sistema pedido en un PAL es necesario expresar las funciones para cada una de las salidas requeridas en formato suma de minterminos. Es por eso que se parte de la tabla de verdad equivalentes a la observada en la Tabla 1. Se obtienen entonces las ecuaciones que se observan en la Tabla 3.

Bit	Expresión
O0	$\bar{E} + In15 + In14 + In13LSB$
O1	$\bar{E} + In15 + In14 + \overline{In13LSB}$
O2	$\bar{E} + In15 + \overline{In14} + In13LSB$
O3	$\bar{E} + In15 + \overline{In14} + \overline{In13LSB}$
O4	$\bar{E} + \overline{In15} + In14 + In13LSB$
O5	$\bar{E} + \overline{In15} + In14 + \overline{In13LSB}$
O6	$\bar{E} + \overline{In15} + \overline{In14} + In13LSB$
O7	$\bar{E} + \overline{In15} + \overline{In14} + \overline{In13LSB}$

TABLA 3: Expresiones lógicas de los bits de salida de la PAL.

Siendo estas expresiones las mismas obtenidas anteriormente. Estas, sin embargo, ya son implementables en un PAL sin ninguna modificación. Dado que en el PAL utilizado (PALCE22V10) cuenta con salidas suficientes

para implementar las funciones anteriormente señaladas se decide no realizar ninguna simplificación adicional.

1.4. Diagrama de tiempos para lectura y escritura

Se presenta en la Figura 6 el diagrama de tiempos tanto como para lectura como para escritura. Cabe aclarar que la única diferencia entre un ciclo de lectura y uno de escritura es que el tiempo que tengo para leer (READ) a los datos es menor debido a que el periférico tiene un tiempo de acceso, dado que el enable tiene una frecuencia fija.

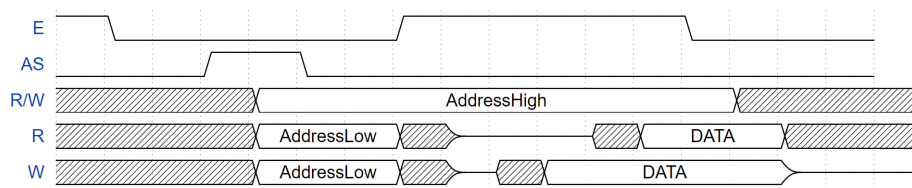


FIGURA 6: Diagrama temporal para lectura/escritura de memoria RAM

2. Ejercicio 2

Se pide diseñar el decodificador de direcciones para el siguiente mapa de memoria, usando un 74LS138.

Address	Dispositivo
C000	ROM 16K
2000	RAM 4K
A000	Puerto de salida de 8 bits
A800	Puerto de entrada de 8 bits

El decodificador 74LS138 es 3-8. Es decir, tiene tres bits de entrada que permiten controlar 8 salidas. Reordenando el mapa de memoria según el Address:

Address	Dispositivo
2000	RAM 4K
A000	Puerto de salida de 8 bits
A800	Puerto de entrada de 8 bits
C000	ROM 16K

A continuación se muestra el espacio que ocupa cada periférico en el mapa de memoria:

- RAM de 4K: Tiene $4 \cdot 1024 = 2^2 \cdot 2^{10} = 2^{12}$ bytes. Por lo tanto, se necesitan 12 bits para poder representar esta cantidad de combinaciones y así poder acceder a cada uno de los bytes de la RAM. Los espacios de memoria dentro de la RAM son de 000 a FFF. Pero dado que dentro del mapa de memoria la RAM arranca en la posición 2000, entonces los espacios destinados a la RAM son de la posición 2000 a la 2FFF.
- Puerto de salida de 8 bits: Está compuesto por un único byte, por lo que con una única posición en el mapa de memoria asignada a este periférico, alcanzaría. La posición que si o si debe estar asignada a este puerto es la A000. Podrían haber más posiciones asignadas a este periférico y se comportarían como espejo de esta.
- Puerto de entrada de 8 bits. Lo mismo que para el puerto de salida de 8 bits, pero debe tener la posición A800 si o si asignada en el mapa de memoria.
- ROM de 16K: Tiene $16 \cdot 1024 = 2^4 \cdot 2^{10} = 2^{14}$ bytes. Por lo tanto se necesitan 14 bits para representar esta cantidad de combinaciones. Las posiciones dentro de la ROM son desde la 0000 hasta la 3FFF. Dado que en el mapa de memoria la ROM comienza en C000, entonces el espacio asignado a la misma va de C000 a FFFF.

Dado que se pide usar el decodificador 74LS138, se tiene en cuenta lo siguiente. El mismo tiene 3 bits de entrada y 8 bits de salida. Con los 3 bits de entrada se puede elegir el periférico, a partir de las 8 combinaciones que este decodificador permite lograr. El bus de direcciones tiene 16 bits: $A_{15} - A_{14} - A_{13} - A_{12} - A_{11} - \dots - A_3 - A_2 - A_1 - A_0$. Si se usan los 3 bits más significativos como entrada al decoder (es decir, $A_{15} - A_{14} - A_{13}$), se obtienen 8 salidas (Y) de las cuales cada una abarca el siguiente rango de posiciones del mapa de memoria:

- Y_0 : 0000 - 1FFF.
- Y_1 : 2000 - 3FFF.
- Y_2 : 4000 - 5FFF.
- Y_3 : 6000 - 7FFF.
- Y_4 : 8000 - 9FFF.
- Y_5 : A000 - BFFF.
- Y_6 : C000 - DFFF.
- Y_7 : E000 - FFFF.

La salida Y_1 permite acceder a las posiciones de memoria de la RAM.

La salida Y_5 en principio permitirá acceder tanto al puerto de salida como al de entrada, por lo que habría que poder distinguir ambos. Dado que el puerto de salida está en la posición A000 y el puerto de entrada en la A800, una opción es usar el bit más significativo del segundo nibble (A_{11}) agregando cierta lógica que permita distinguir entre uno y el otro. Considerando que las salidas del decoder son activo bajas y que los chip select de los periféricos también, se emplea la siguiente tabla de verdad para armar la lógica necesaria de menor complejidad.

Y_5	A_{11}	CS Puerto de salida	CS Puerto de entrada
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	1

Con mapas de Karnaugh se llega a que el CS del puerto de salida se relaciona con Y_5 y A_{11} por medio de: $CS = Y_5 + A_{11}$, mientras que para el caso del puerto de entrada: $CS = Y_5 + !A_{11}$, recordando que el CS es activo bajo.

Las posiciones que controlan la ROM son abarcadas tanto por Y_6 como por Y_7 . Por lo tanto habría que unir ambas salidas, para que las dos vayan al CS de la ROM. Para agregar estas cosas, hay que considerar que las salidas del decoder son activo-bajo. La tabla de verdad que representa esta situación es la siguiente:

Y_6	Y_7	CS de la ROM
0	0	0
0	1	0
1	0	0
1	1	1

Esto es lo mismo que decir que el CS de la ROM está dado por: $CS = Y_6 \cdot Y_7$.

Así es como se obtiene el siguiente esquema del decoder:

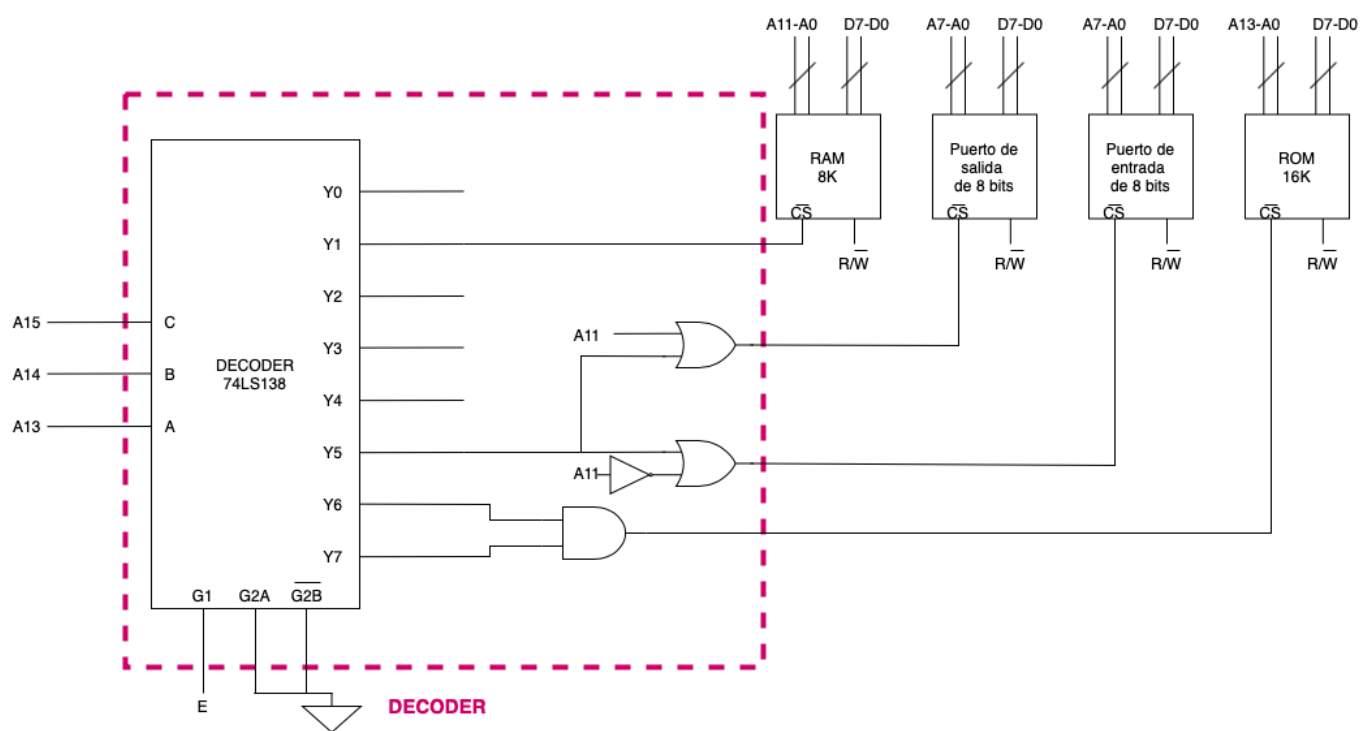


FIGURA 7: Esquemático completo del decoder.

3. Ejercicio 3

Todo microprocesador que se conecta mediante el bus a sus periféricos, tiene un número límite de dispositivos a los que se puede conectar a la salida de dicho bus, esto se debe a que en el bus de conexión presenta diferentes buffers en cada uno de sus bites, estos buffers presentan impedancias capacitivas en su entrada, por lo que se comportan como si tuvieran capacitores conectados a tierra en su entrada.

Estas capacitancias pertenecientes a los buffers del bus se deben a los transistores de la etapa de entrada interna del buffer, a continuación en la figura 8 se muestra un esquema simplificado del capacitor que se puede considerar en cada bit del bus.

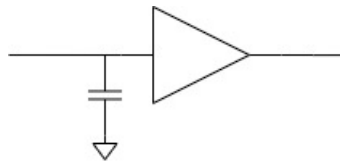


FIGURA 8: Circuito simplificado para cada bite

Se puede ver entonces que en cada línea del bus existirá un capacitor conectado a tierra producto de cada uno de los buffers de cada bit, por ello se entender como un solo buffer de 8 entradas, con la capacidad correspondiente asociada, de esta manera es mas fácil comprender como luego la conexión de un mayor numero de periféricos modifica la capacidad total de entrada del sistema.

Como se puede ver en la figura 9, al aumentar el número de periféricos se agregan los buffers correspondientes a cada uno, esto añade una capacidad equivalente en cada uno de los bits, por lo que al conectar todos en paralelo (haciendo uso del mismo bus) se obtiene una capacidad final que resulta de la suma de las capacidades de cada uno.

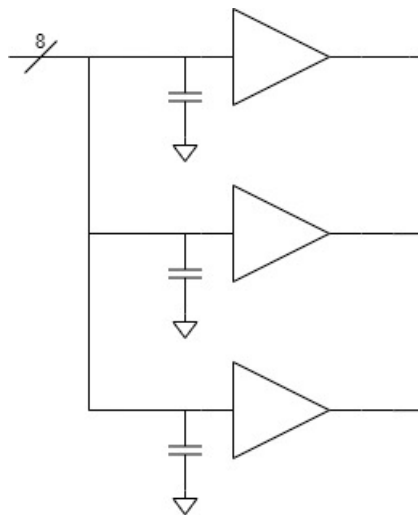


FIGURA 9: Circuito simplificado de las capacidades asociadas a la conexión de más de un puerto al bus de datos.

El hecho de tener capacidades limita la frecuencia de clock permitida para que el circuito se comporte como es esperado y no entregue o mida valores erróneos en los periféricos conectados. Por lo que para una frecuencia

de trabajo dada, existe un numero máximo de periféricos que se pueden conectar en la salida, si se conectaran mas de este número, la capacidad resultante aumentaría provocando que el circuito RC que se forma presente una respuesta más lenta lo que perjudicaría el envío o recibo de la información entre en puerto y la CPU.

Una forma de solucionar este problema puede realizarse en base a la conexión de un repetidor. Este se constituye de 2 buffers (para permitir el envío en ambos sentidos de la información), ambos buffers tristate están controlados por las líneas R/\bar{W} y aquella que corresponda a la información de la zona de la memoria donde se trabajará para ese repetidor. Esto se puede ver en la figura 15.

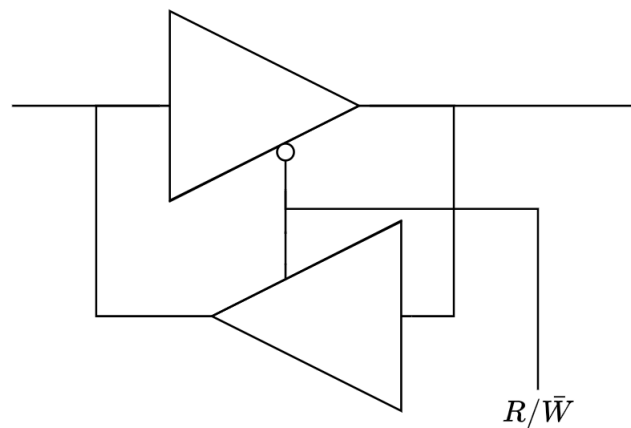


FIGURA 10: Esquema representativo del repetidor que se podría implementar como solución a la capacidad total de valor elevado.

De esta manera se podría dividir en partes la cantidad total de dispositivos es decir se podría por ejemplo dividir la cantidad total de dispositivos en 2, un conjunto de dispositivos antes y el otro luego del repetidor, de esta manera no se suman las capacidades de ambos grupos como ocurriría sin el repetidor. Se puede de esta manera agregar un conjunto de repetidores con el fin de reducir la capacidad equivalente total que se termina viendo desde la CPU evitando el problema.

4. Ejercicio 4

En este ejercicio se detallarán algunas de las formas de interconexión de sistemas TTL que trabajan con lógica de 5V con otros que trabajan con lógica TTL de 3,3V.

Aquellos que trabajan con lógica de 5v se los suele denominar 5V-TTL, y los que implementan tecnologías de 3.3V TTL se los suele denominar LVTTL.

Las distintas tecnologías de trabajo implementan niveles mínimos y máximos que son considerados ceros o unos por el sistema. Estos son los que constituyen los niveles de trabajo del mismo. En el medio entre la zona de high y la zona de low se encuentra la zona gris en la cual el sistema no puede distinguir si la entrada constituye un cero o un uno lógico. Esto se puede apreciar en la figura ??.

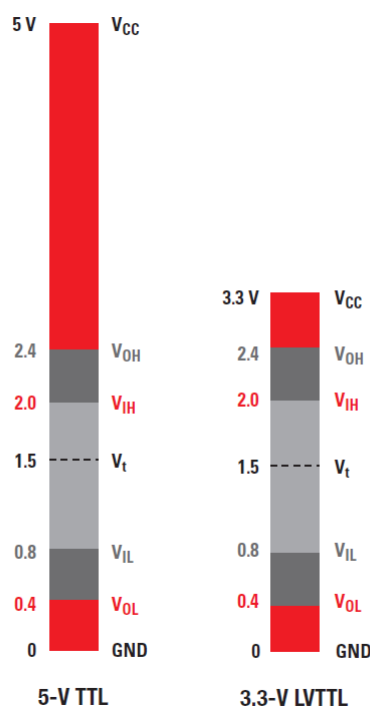


FIGURA 11: Esquema de los niveles de tensión presentes en tecnologías que implementan 5v-TTL y LVTTL.

Se puede ver en la imagen de la figura 15 que para cada una de las tecnologías existen diferentes indicadores, los mismos son:

GND: Mínima tensión permitida para salida o entrada de señal Low.

V_{OL} : Máxima tensión permitida para la salida de tensión en Low.

V_{IL} : Máxima tensión permitida para la entrada de tensión que se considera Low.

V_{IH} : Mínima tensión permitida para la entrada que se considera High.

V_{OH} : Mínima tensión permitida para la salida en High.

De esta manera quedan las zonas de trabajo que se permiten para la entrada y salida de la tecnología en cuestión. Estas zonas se encuentran en la figura 15 coloreadas de diferentes colores. Color rojo para la zona de las tensiones de salida, zona gris oscura mas la zona roja constituyen la zona de tensiones de entrada. Luego la zona gris clara constituye la zona intermedia en la que el sistema no puede distinguir el estado.

Se puede ver que tanto la tecnología 5V-TTL como la tecnología 3.3V-LVTTL tienen los mismos valores a excepción de la tensión máxima de salida que en un caso es de 5V y en el otro de 3V.

Por esta razón se puede entender que la conexión que se realice en un sólo sentido entre un sistema de 3.3V-LVTTL y otro de tecnología 5V-TTL se puede realizar sin ningún tipo de circuito, es decir con un conductor simple funcionaría correctamente ya que cumple con las condiciones de conexionado, es decir que las zonas de salida tanto para Low como High están dentro de las zonas de los valores de tensión permisibles de entrada del sistema.

No ocurre lo mismo si se quisieran conectar los dispositivos en el sentido inverso, es decir conectar la salida del sistema 5V-TTL con la entrada del sistema 3.3V-LVTTL. Ya que en este caso la conexión no es viable sin algún tipo de circuito que permita realizar una adaptación de los niveles de tensión de ambos. Este elemento es necesario ya que la salida del sistema de 5V puede tener valores superiores a los 3.3V de tensión máxima que permite el otro sistema.

Una conexión simple que permite realizar la conexión unilateral de estos sistemas puede consistir en 2 resistencias que conformen un divisor resistivo, para ello posibles valores comerciales que se podrían emplear serían: 1.8k y 3.3k, este juego de resistencias comerciales permite tener una tensión de salida muy similar a la requerida: 3.3V de tensión máxima. Pero esto no sería correcto ya que podría pasar que la tensión de salida de la primera compuerta sea 2.4 V (correspondiente al mínimo permitido para la tensión de High de salida) lo cual por el divisor resistivo daría un valor de 1.55V, quedando en la zona gris clara es decir el otro sistema no lo identificaría como 1 lógico aun cuando debería ser así. Por lo tanto hay que encontrar otra solución.

Una posible solución se puede implementar mediante un transistor mosfet y 2 resistencias de pull-up. A continuación se muestra la forma de funcionar del posible circuito implementado de esta manera.

Para la explicación del circuito se tomarán los 4 modos de funcionamiento del mismo de forma separada con imágenes para cada caso¹:

El primer modo de funcionamiento considerado es cuando el sistema de 3.3V transmite un uno lógico (High) hacia el sistema de tecnología 5V. En ese caso la tensión entre gate y source es 0 (ambos están a 3.3V). De esta manera el mosfet se comporta como un circuito abierto. De esta manera la resistencia de pull up conectada en la sección de 5V brinda la tensión High requerida.

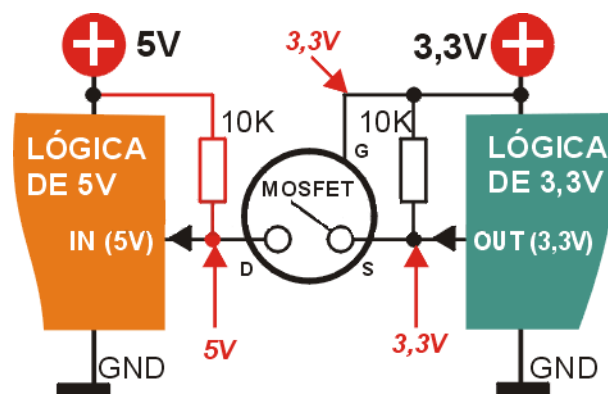


FIGURA 12: Esquema básico de un circuito para comunicación de 1 lógico de 3.3v a 5v.

El segundo modo de funcionamiento ocurre cuando el dispositivo de 3.3v transmite un cero lógico (low) al

¹Imágenes tomadas del sitio web <https://www.inventable.eu/>

sistema de 5V. En este caso la tensión de salida del sistema de 3.3V es 0 por lo que ahora la tensión entre gate y source es de 3.3V, esto hace que el transistor conduzca comportándose como un circuito cerrado. Esto hace que la entrada de 5v sea cero y que la resistencia de pull-up disipe la tensión de 5V en ella, por lo que se consigue la entrada de cero volts en el sistema de 5v.

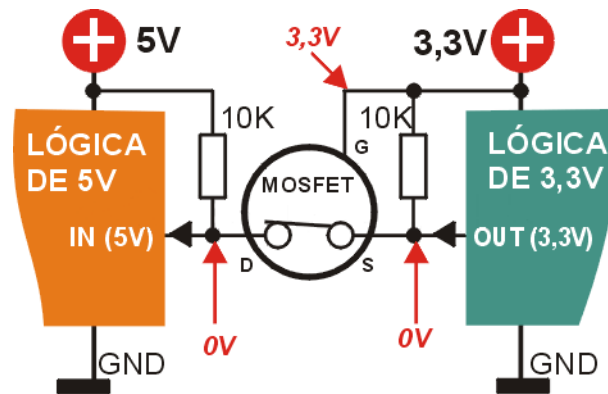


FIGURA 13: Esquema básico de un circuito para comunicación de 0 lógico de 3.3v a 5v.

El tercer modo de funcionamiento ocurre cuando el sistema que trabaja a 5v ha de emitir una señal de salida de valor 0v, en esta condición el transistor se comporta como un diodo con una caída de tensión asociada del orden de 0,7V, esta será la tensión que se verá desde la otra compuerta, esto no genera problemas debido a que se encuentra dentro del rango permisible para las señales que se han de considerar un cero lógico, por lo tanto el circuito se comporta como es debido.

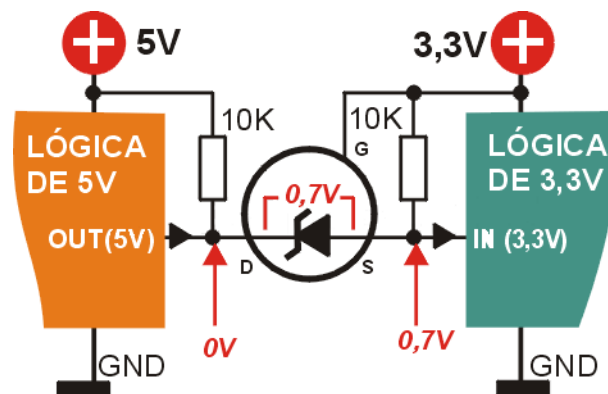


FIGURA 14: Esquema básico de un circuito para comunicación de 1 lógico de 5v a 3.3v.

El cuarto y último modo de funcionamiento del circuito ocurre cuando el sistema de 5V transmite un uno lógico (High) y el sistema de 3.3V es el que lo recibe. En este caso el sistema de 5V esta forzando a la salida un uno lógico, 5v, por lo que el drain del mosfet esta a 5v y no conduce, entonces en el sistema de 3.3v la entrada del mismo presenta 3.3v ya que esa tensión es transmitida mediante la resistencia de pull-up (por donde circula poca corriente), obteniéndose el uno lógico en el sistema de 3.3v correspondiente.

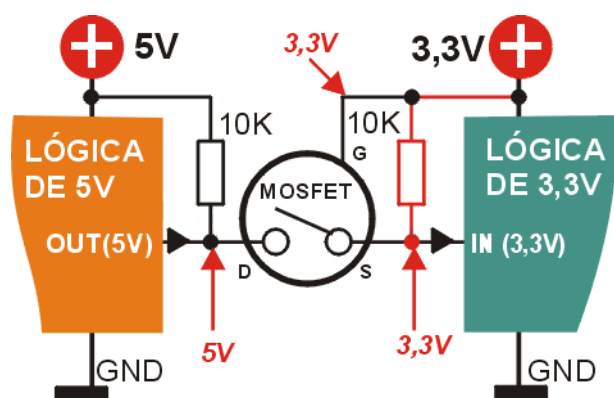


FIGURA 15: Esquema básico de un circuito para comunicación de cero lógico de 5v a 3.3v.

5. Ejercicio 5

Según el programa de la tabla 4 se pide realizar el diagrama de tiempos, tomando de referencia el manual de referencia del HC11 .

```
org    $C000
ldaa   #$A5
L1:    staa $4000
       jmp  L1
```

TABLA 4: Programa a analizar

Del manual de referencnia del microprocesador se pueden obtener las especificaciones del comportamiento de los buses de interes para cada instruccion ejecutada en el programa, lo cual se muestra en la tabla 5.

ldaa			staa			jmp		
Add	Data	R/ \overline{W}	Add	Data	R/ \overline{W}	Add	Data	R/ \overline{W}
OP	86	1	OP	B7	1	OP	7E	1
OP+1	ii	1	OP+1	hh	1	OP+1	hh	1
			OP+2	ll	1	OP+2	ll	1
			hhl	(A)	1			

TABLA 5: Valores del bus de datos y direcciones para cada instruccion.

De esta forma se puede observar que el R/\overline{W} siempre se encuentra en 1 según la información tomada de la cartilla de referencia del HC11, por lo tanto no se lo incluirá en el diagrama de tiempos.

De esta forma es posible realizar el diagrama de tiempos como se muestra en la figura 16. En el mismo la flecha azul indica que el diagrama continua siguiendo la misma secuencia que se observa a partir de donde esta indicado.

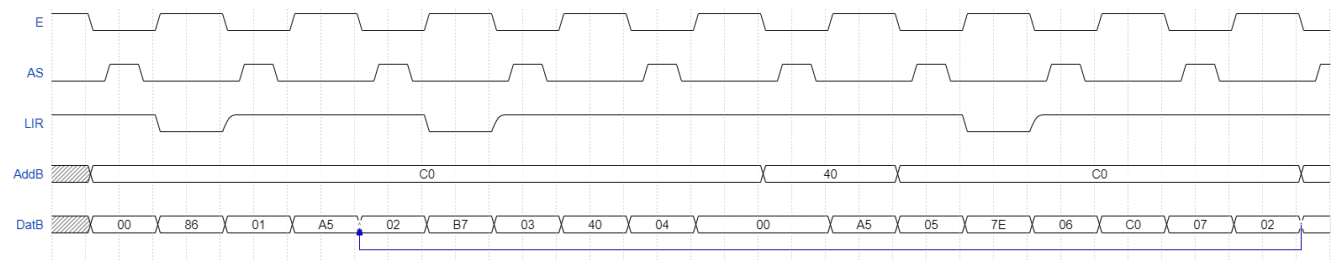


FIGURA 16: Diagrama de tiempos