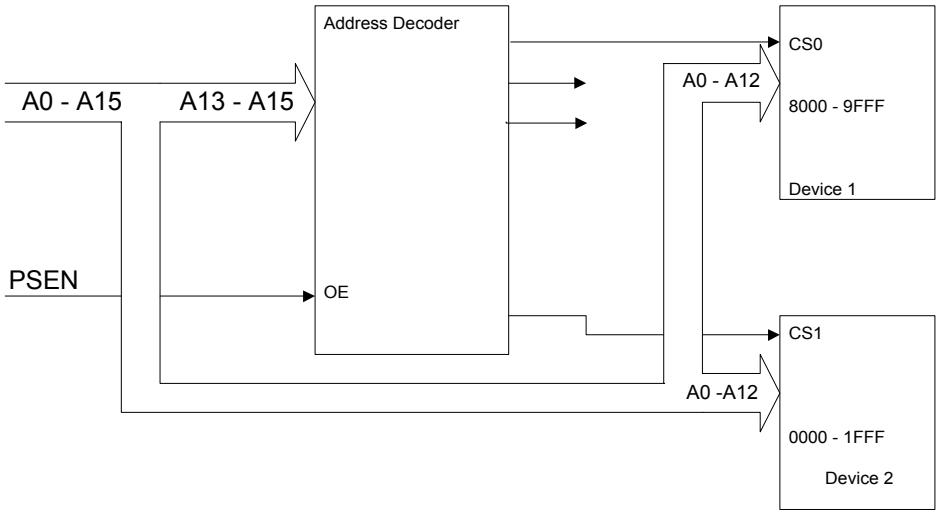
Address Decoding

We have looked at the lines provided by the 8051 to interface to external devices. Now lets look at the actual address decoder.

Address Decoder:

Computer logic that will use address and timing information from the processor to generate Chip Select (CS) lines used to enable external devices.



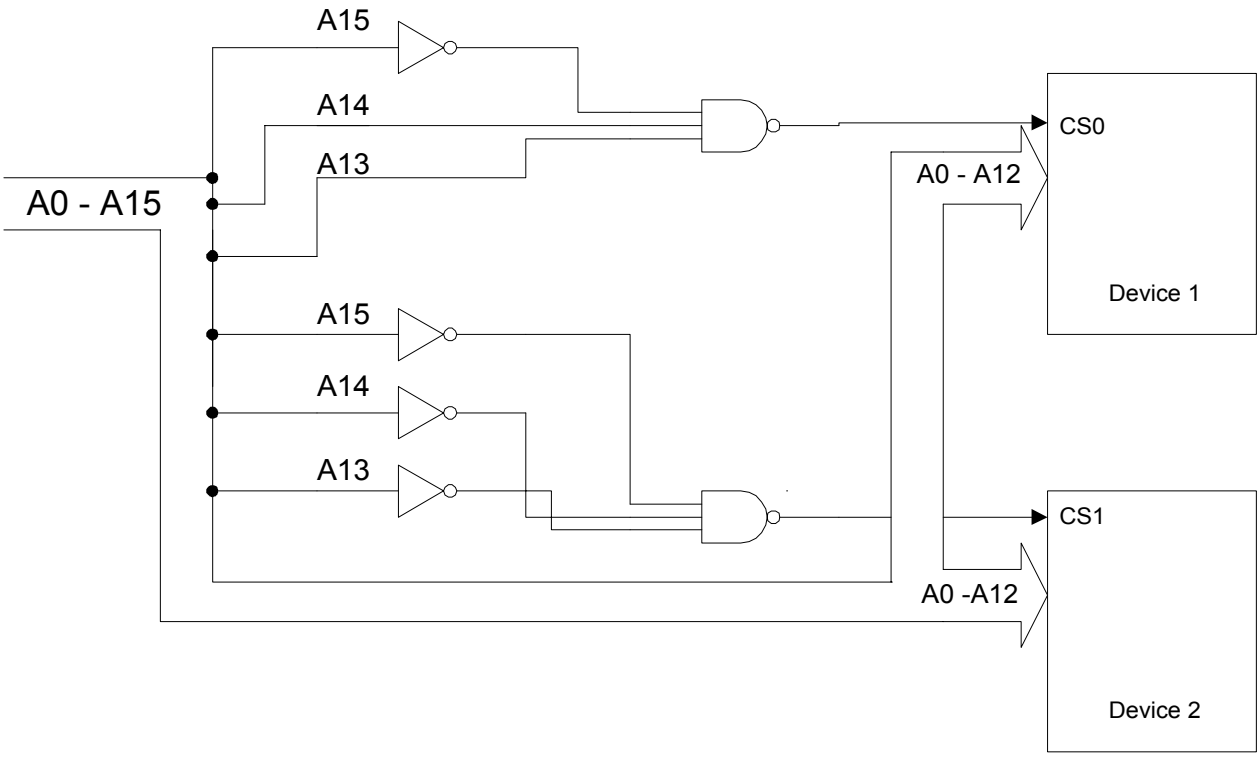
To design an address decoder to provide a chip select signal for any device connected to the bus, follow the instructions below:

A. Address Table: Draw an address table showing the address range for each device to be decoded. The table must also show the contents of each bit of the address range. If a bit must be high to address the device denote it with a "1" in that address bit position. If a bit must be zero to address the device, denote it with a "0" in that address bit position. If the bit can either be a "1" or a "0" then denote this with an "X". Use the above diagram as an example.

	Add. Range	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Device 1	8000 – 9FFF	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
Device 2	0000 – 1FFF	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X

B. Select Address Lines: From the table the address lines with either a "1" or a "0" will be used to generate the chip select signal. The address lines with an "X" in their box will go directly from the processor to the device. These lines will be used by the device to select either the memory cell desired or to set an internal register in the device. By looking at the table it can be seen that address lines A15, A14, and A13 need to be decoded to generate a

unique chip select line for each device. By setting A15 = 1. A14 = 0, and A13 = 0, device 1 will be selected. Similarly, setting A15, A14, and A13 to 0 will select device 2. In the above example we can simplify the design by setting A15 = 1 for device 1 and A15 to 0 to select device 2.



Using discrete gates to decode the chip select lines can be the most cost effective solution if you are small, dedicated system. For larger applications requiring many peripheral devices (requiring many chip select lines) it is best to use an LSI device like the 74ls138 shown below.

FAIRCHILD
SEMICONDUCTOR

February 1984
Revised February 1999

MM74HCT138
3-to-8 Line Decoder

General Description

The MM74HCT138 decoder utilizes advanced silicon-gate CMOS technology, and are well suited to memory address decoding or data routing applications. Both circuits feature high noise immunity and low power consumption usually associated with CMOS circuitry, yet have speeds comparable to low power Schottky TTL logic.

The MM74HCT138 have 3 binary select inputs (A, B, and C). If the device is enabled these inputs determine which one of the eight normally HIGH outputs will go LOW. Two active LOW and one active HIGH enables (G1, G2A and G2B) are provided to ease the cascading decoders.

The decoders' output can drive 10 low power Schottky TTL equivalent loads and are functionally and pin equivalent to the 74LS138. All inputs are protected from damage due to static discharge by diodes to V_{CC} and ground.

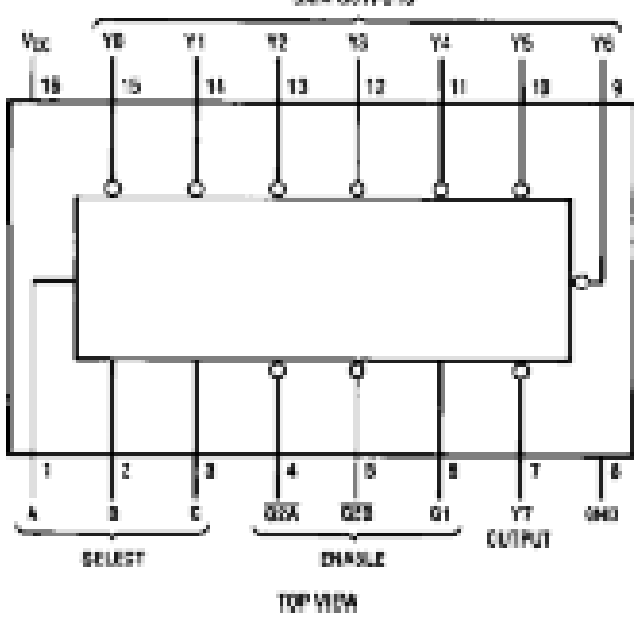
MM74HCT devices are intended to interface between TTL and NMOS components and standard CMOS devices. These parts are also plug-in replacements for LS-TTL devices and can be used to reduce power consumption in existing designs.

Features

- TTL input compatible
- Typical propagation delay: 20 ns
- Low quiescent current: 80 µA maximum (74HCT Series)
- Low input current: 1 µA maximum
- Fanout of 10 LS-TTL loads

Connection Diagram

Pin Assignments for DIP, SOIC, SOP and TSSOP



Truth Table

Inputs					Outputs							
Enable		Select										
G1	$\overline{G2}$ (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care
Note 1: $\overline{G2} = \overline{G2A} + \overline{G2B}$

Designing a chip select decoder circuit around the 74LS148 simplifies the circuit while providing eight or more unique chip selects. In addition to providing the address inputs (select A, B, and C) it also provides enable lines G1 and G2. This are used to enable the output and would typically be tied to the PSEN line from the 8051. An example of designing with the 74LS138 is shown below.

Ex. Design the chip select circuit to decode the following:

ROM 1	F000 – FFFF
ROM 2	E000- EFFF
RAM 1	0000 – 1FFF
RAM 2	2000 – 3FFF
PIA 1	8000 – 8003
UART 1	A000 – A007

Step 1 Fill in the address table showing each device and the lines necessary to make that device function.

	Add. Range	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM 1	F000 – FFFF	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X
ROM 2	E000 – EFFF	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
RAM 1	0000 – 1FFF	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
RAM 2	2000 – 3FFF	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X
PIA 1	8000 – 8003	1	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
UART 1	A000 – A007	1	0	1	0	0	0	0	0	0	0	0	0	0	X	X	X

Step 2 Looking at the table, determine which address lines are required to uniquely select any one of the six devices. If we use A15, A14, and A13, we can uniquely select all of the devices except the two ROMs. For these we will have to include A12.

Step 3 Draw the schematic.