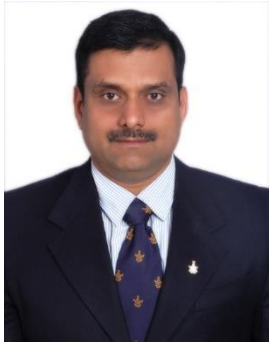
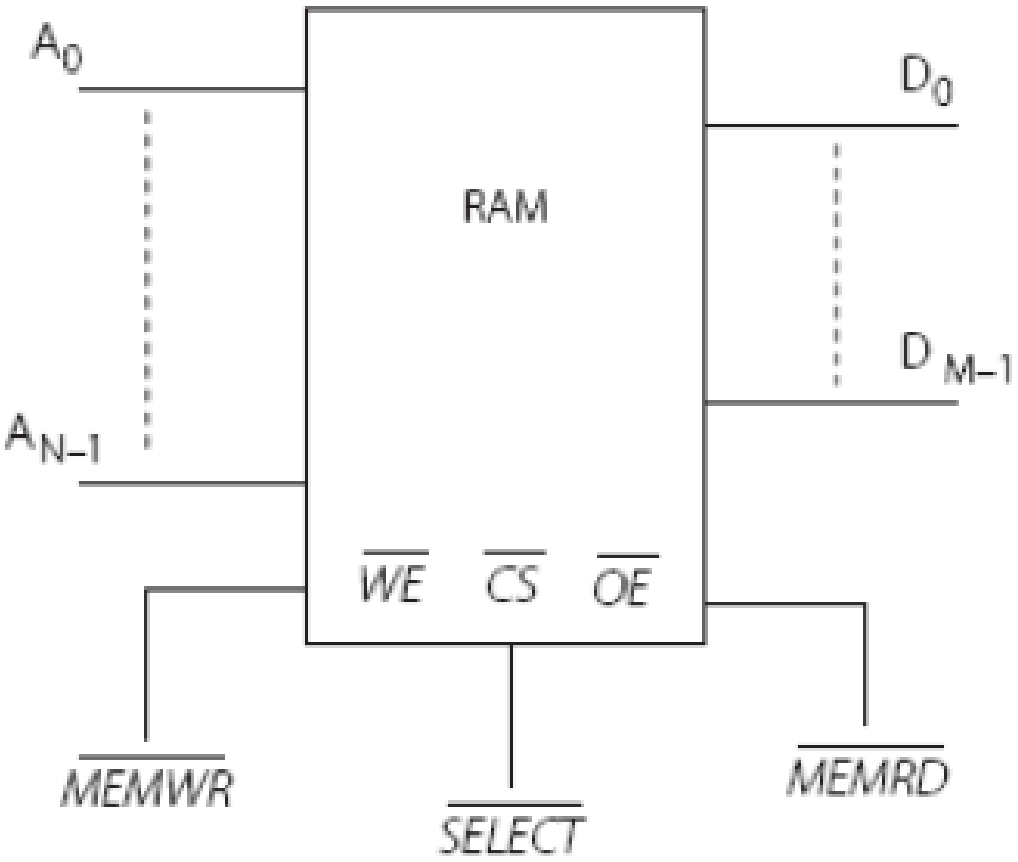
Microprocessors and Interfaces: 2021-22
Lecture 22 :
Memory Interface
Part 1 : Address Decoding

By Dr. Sanjay Vidhyadharan

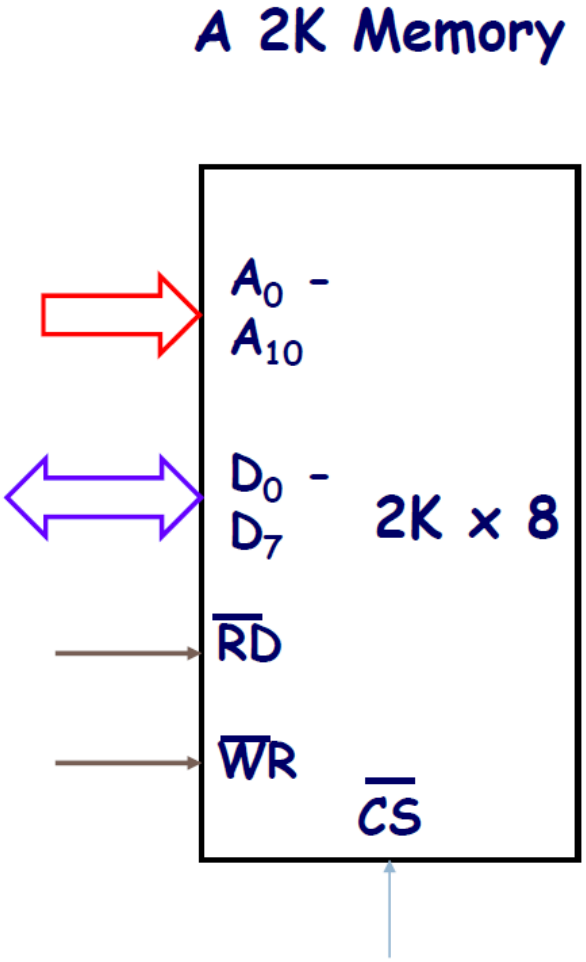


MEMORY



Sar ran

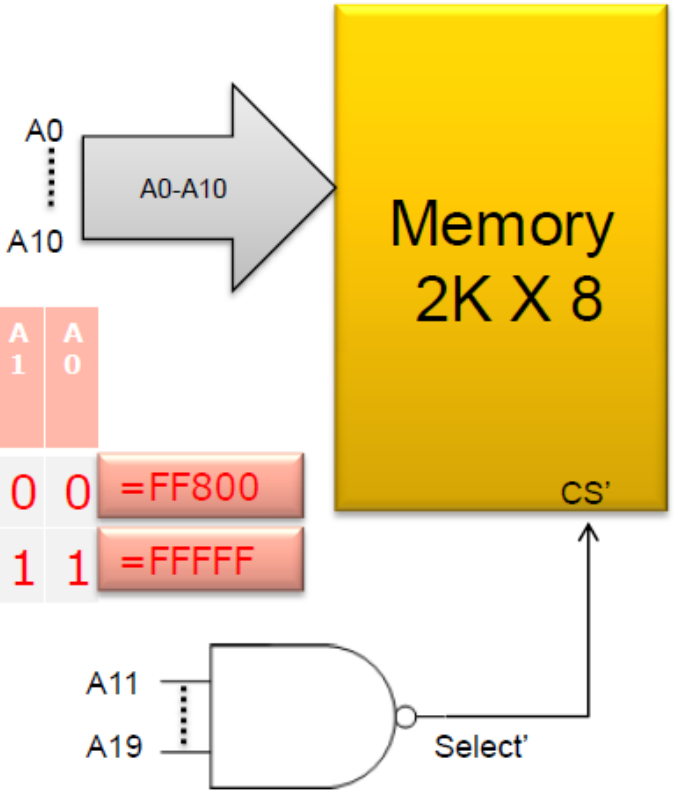
Memory Chip



Sanjay Kharan

Address Decoding

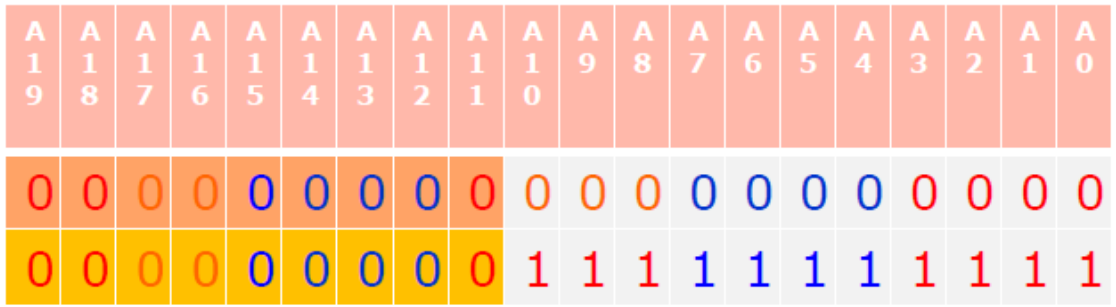
```
1111 1111 1XXX XXXX XXXX

          or

1111 1111 1000 0000 0000 = FF800H
          to
1111 1111 1111 1111 1111 = FFFFFH
```



A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	=FF800
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	=FFFFF

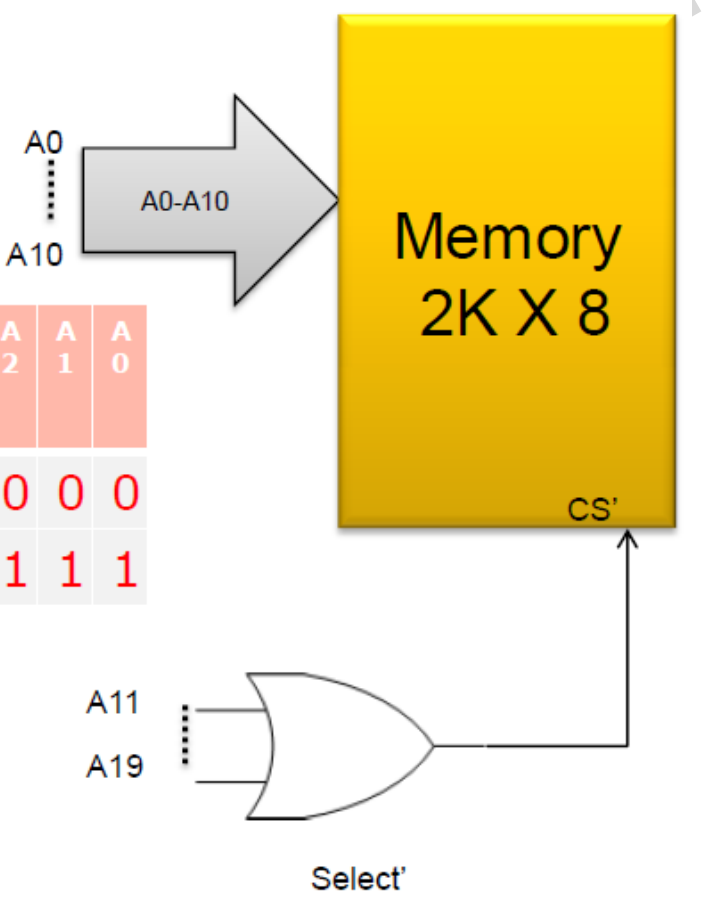
Address Decoding

Design the same memory with address range starting from 00000



A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Range= 00000 -- 007FF



Address Decoding

- Can be built using:
- Logic gates
- Block decoders (e.g. 2 × 4, 3 × 8 …)
- Programmable logic (PLAs,PLDs, FPGAs …)

Sanjay V.

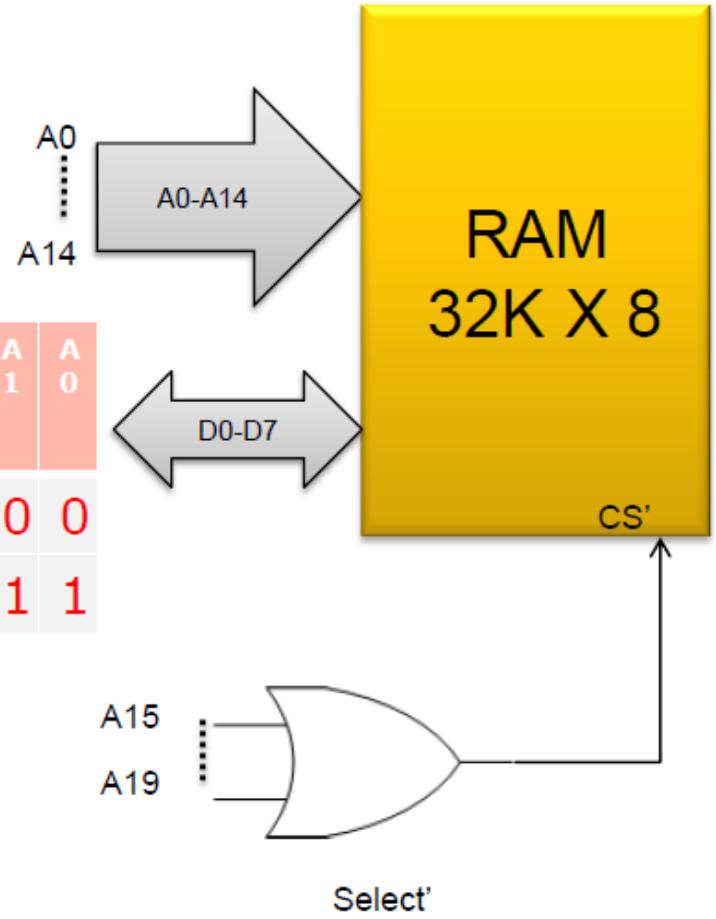
ELECTRICAL ELECTRONICS COMMUNICATION INSTRUMENTATION

Example

Design an address decoder for a 32 K × 8 RAM.

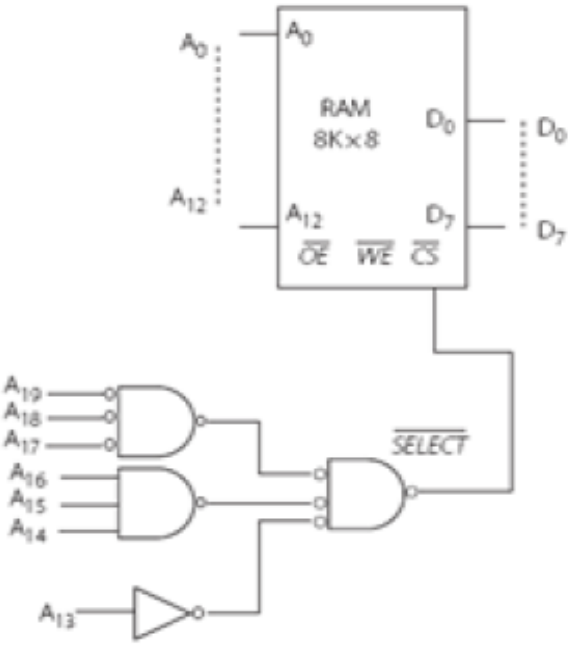
A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

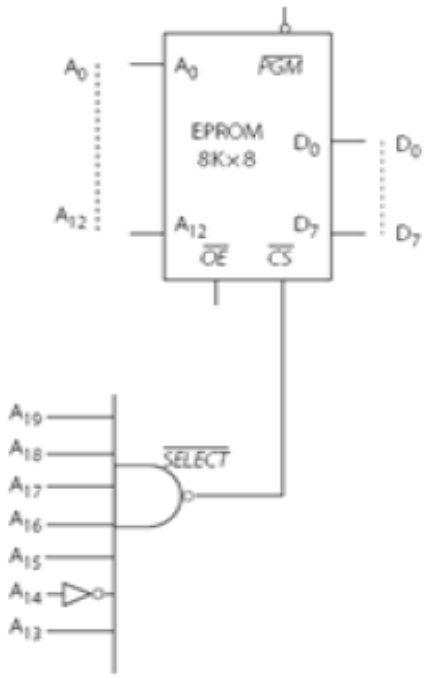
Range= 00000 -- 07FFF



Example

Design an address decoder for a 8K RAM from 1E000 and 8K ROM from FA000

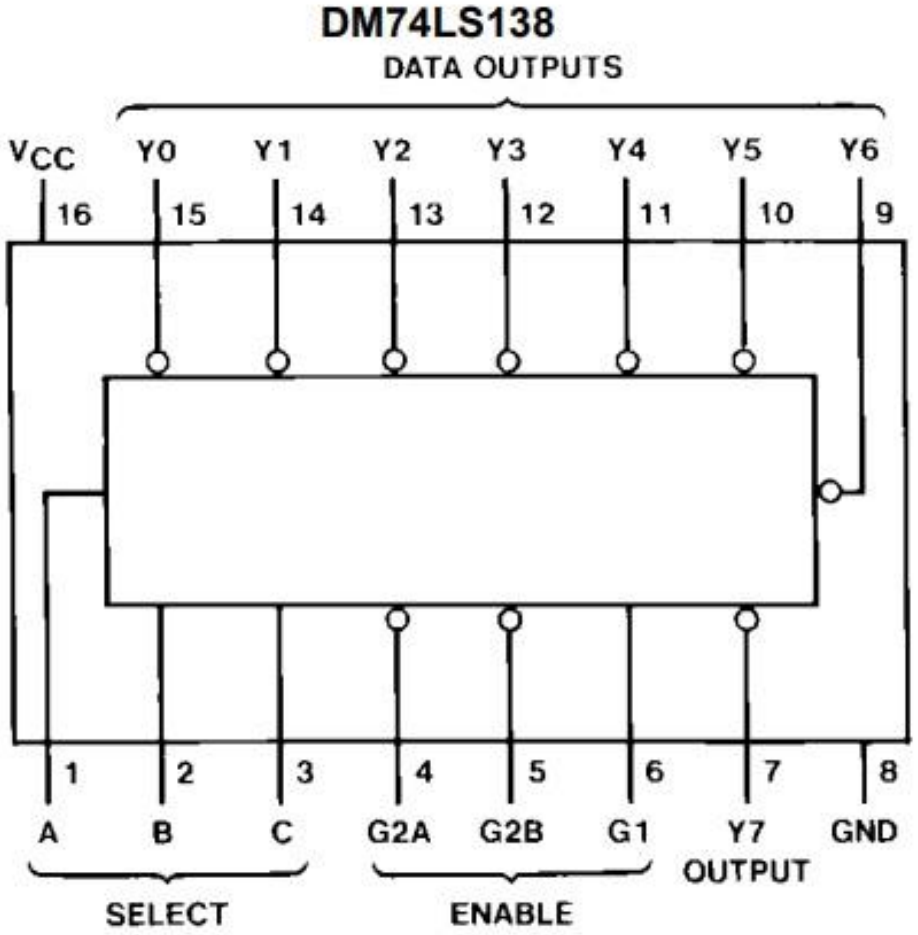




	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAM low	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1E000
RAM high	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFF
ROM low	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	FA000
ROM high	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FBFFF

Using Block Decoders

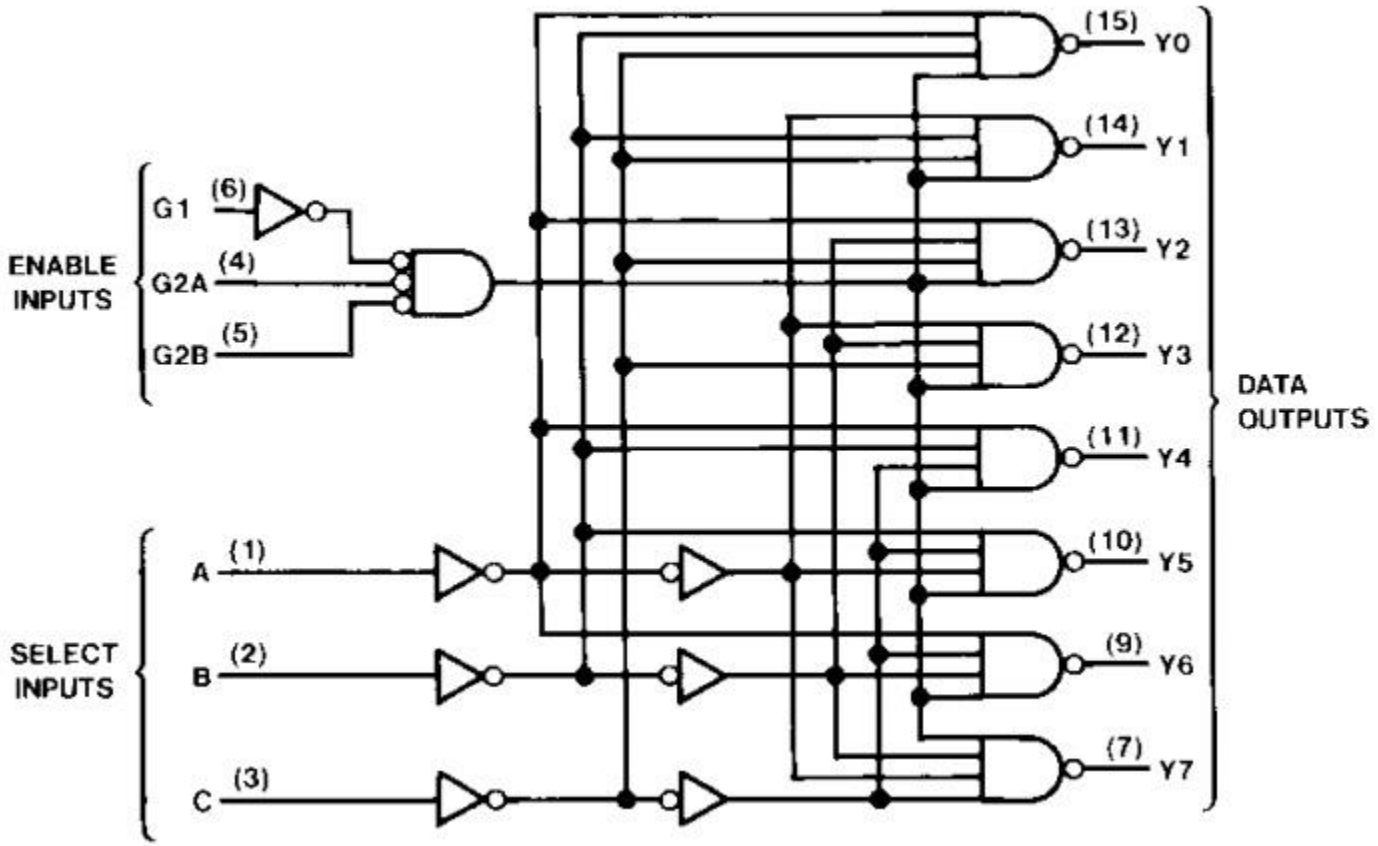
SN54/74LS138 Decoder/Demultiplexer



Jayaharan

Using Block Decoders

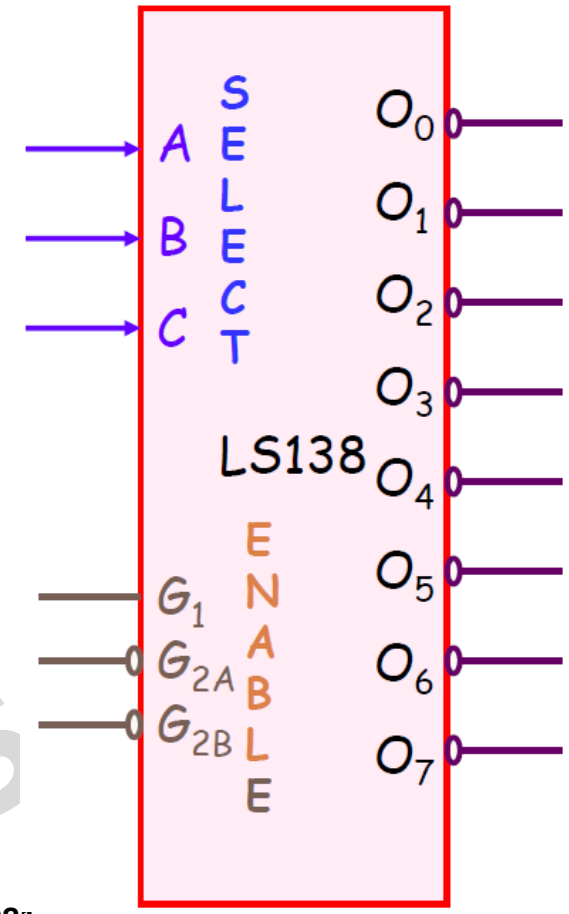
SN54/74LS138 Decoder/Demultiplexer



ELECTRICAL ELECTRONICS COMMUNICATION INSTRUMENTATION

Using Block Decoders

SN54/74LS138 Decoder/Demultiplexer



INPUT						OUTPUT							
ENABLE			SELECT										
G_1	G_{2A}	G_{2B}	A	B	C	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

Using Block Decoders

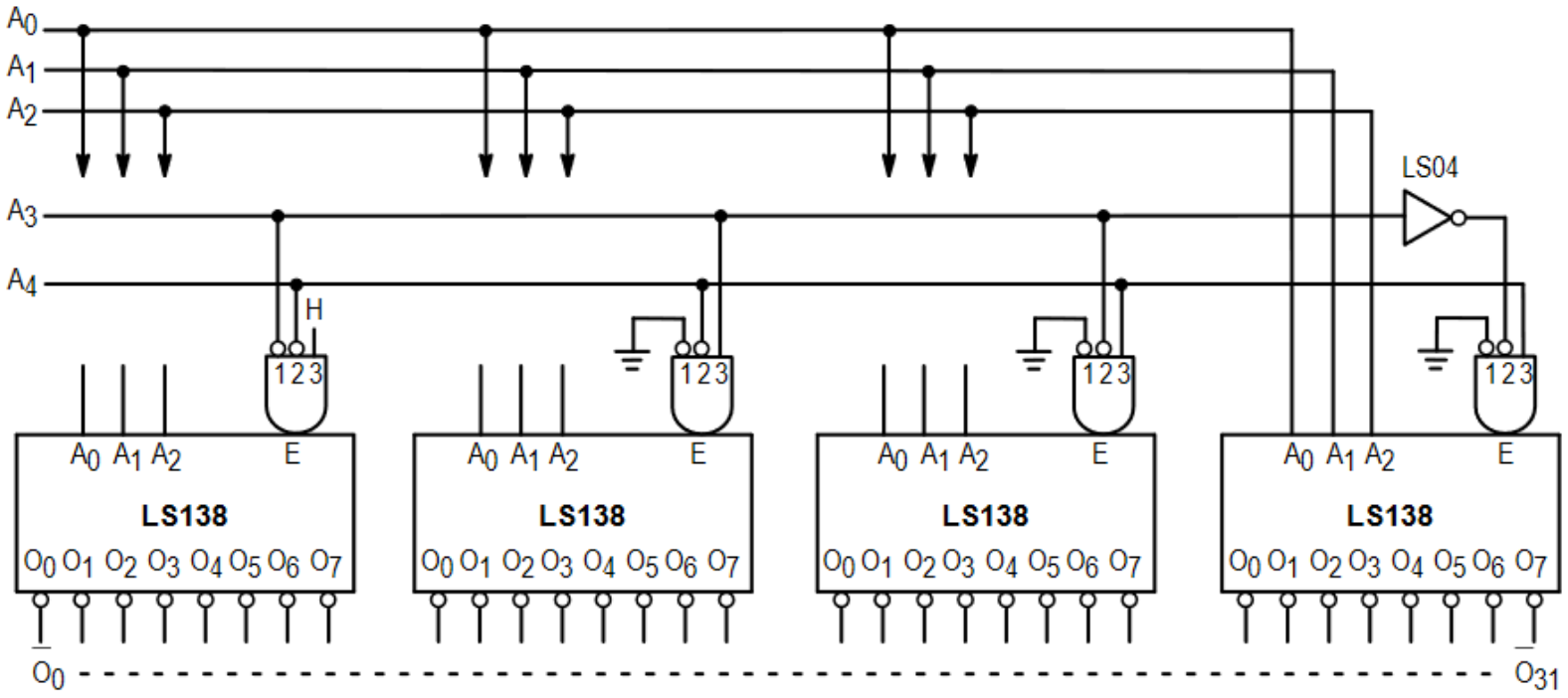
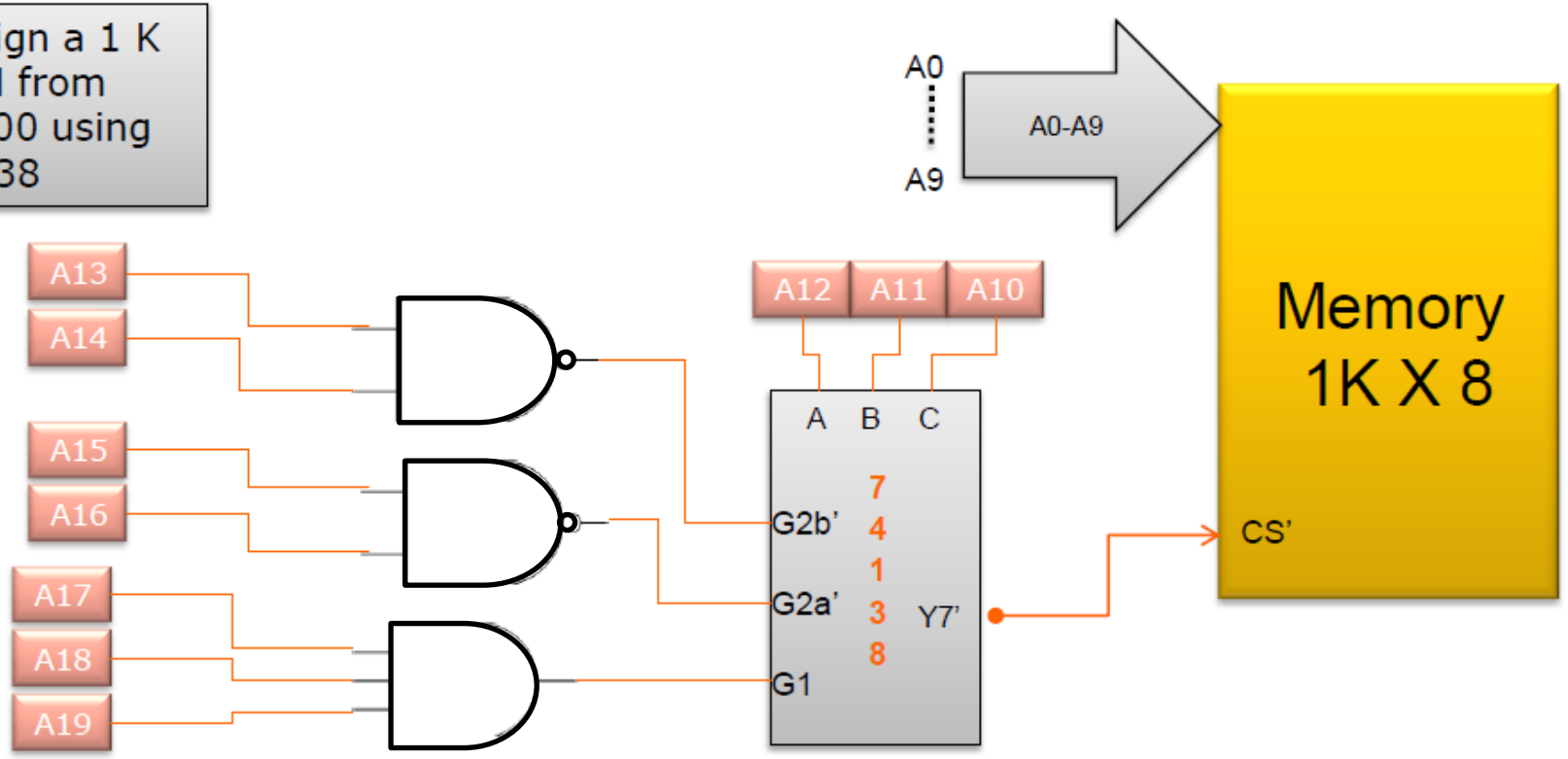


Figure a

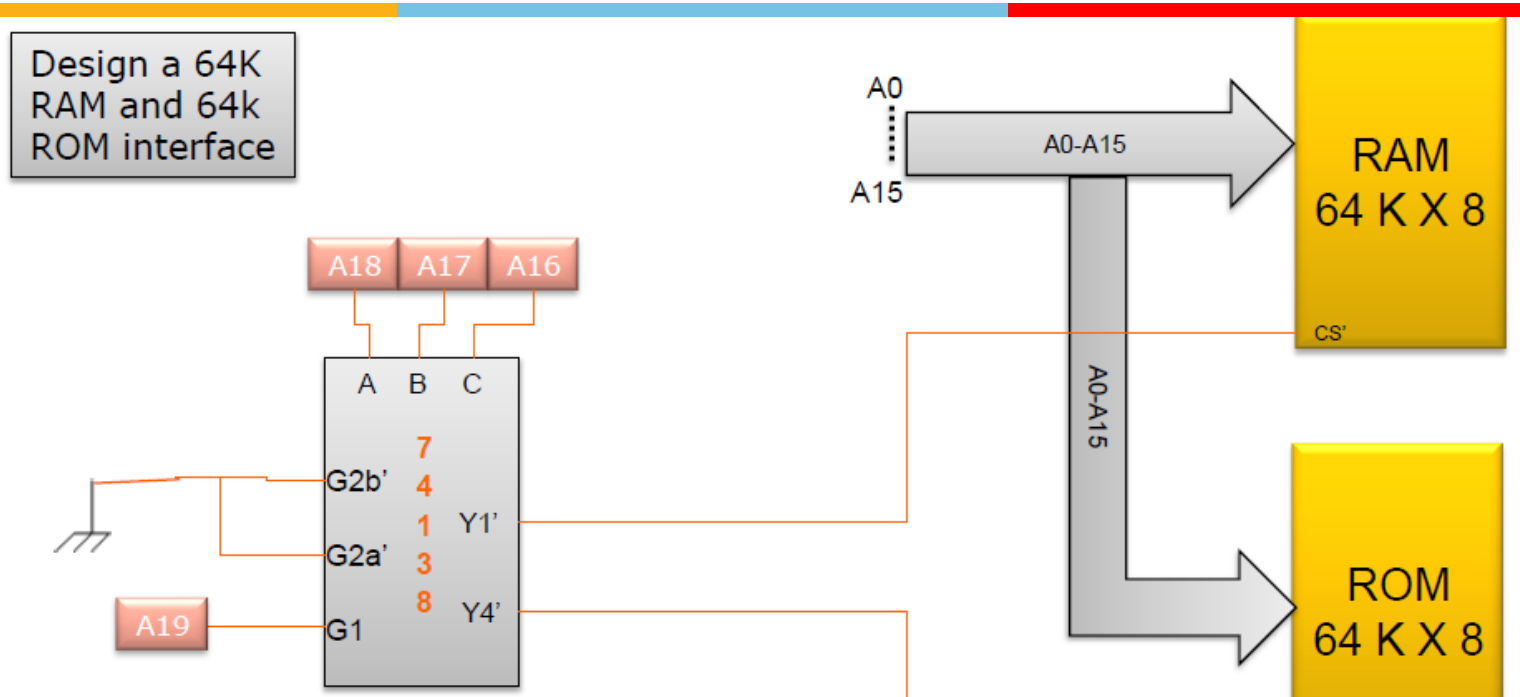
Example

Design a 1 K RAM from FFC00 using 74138



	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAM low	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	FFC00
RAM high	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFF

RAM & ROM using a Decoder



	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
RAM low	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C0000
RAM high	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	CFFFF
ROM low	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90000
ROM high	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	9FFFF

Thank You

Sanjay Vidhyadharan