Memory Interfacing

ECE 511: Digital System & Microprocessor

What we will learn in this session:

- Review several important aspects of memory access.
- Types of memory.
- How to design a Memory Address Decoder:
  - Full addressing.
  - Partial addressing.
  - Using 74LS138 and 74LS139 decoders.

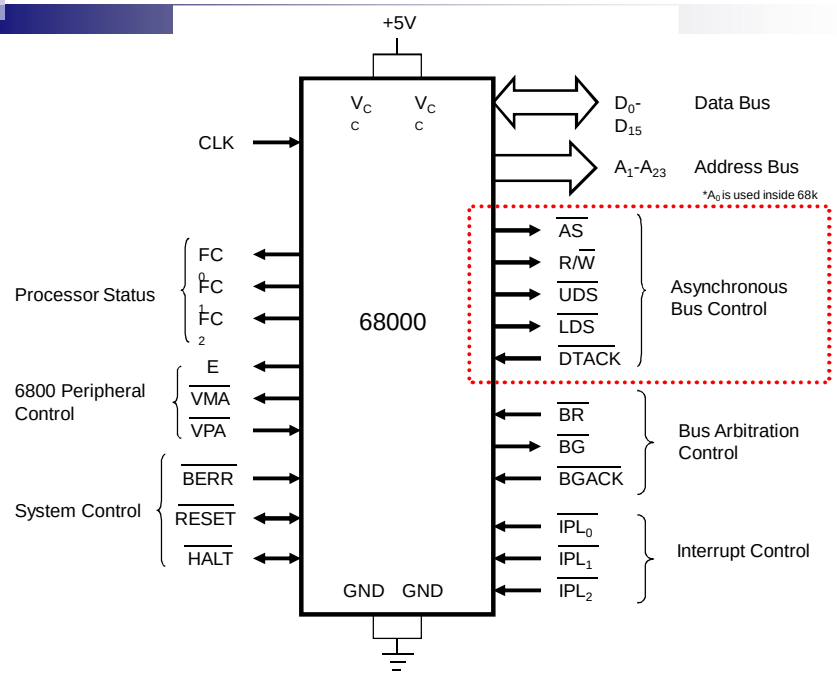
M68k & Memory

M68k & Memory

- M68k has limited space to store data & instructions:
  - 8 x Data registers.
  - 1 x Instruction Register.
- Not enough for practical applications.
- Memory expands storage space:
  - 16 MB maximum space.
  - Stores instructions & data.

M68k & Memory

- M68k has 24-bit (23 + 1) address bus:
  - Can address 2^{24} locations.
  - 16,777,216 locations (16 MB).
  - Can store much more data, instructions.



$\overline{AS}$ – Address Strobe

- Purpose:
  - Indicates that M68k is exclusively using system bus.
- Activated when M68k wants to use system bus:
  - Reading from memory.
  - Writing to memory.
  - Access peripherals.

R/$\overline{W}$

- Used to specify read/write operation.
- 1 pin, output.
- Three states:
  - High (1): read (default).
  - Low (0): write.

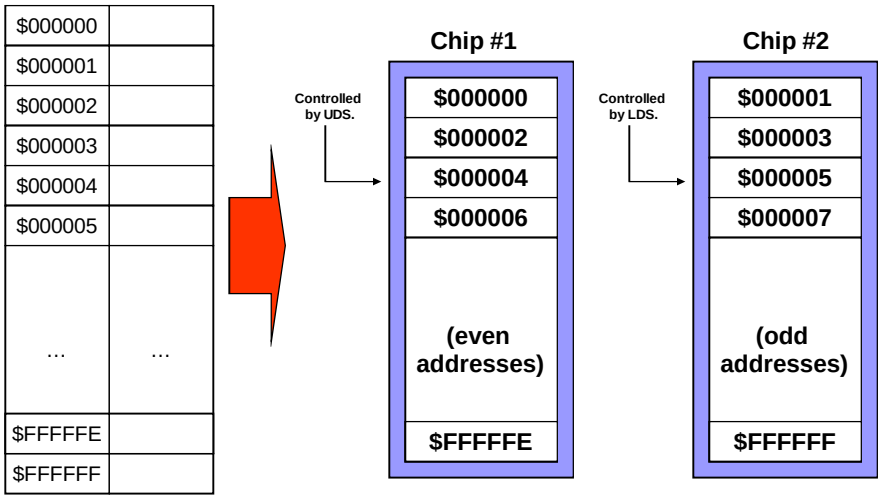
$\overline{\text{DTACK}}$ – Data Transfer Acknowledge

- Indicates device ready to begin data transfer.
- Generated by external device being accessed.
- When M68k receives signal, knows data transfer can be started, begins read/write operation.
- During data transfer, M68k inserts wait states until $\overline{\text{DTACK}}$ is received.

$\overline{\text{UDS}}$/$\overline{\text{LDS}}$

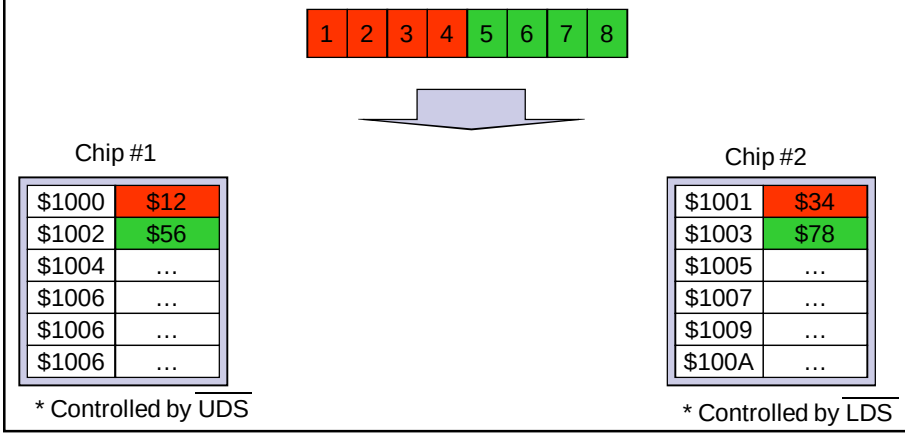
- Used to activate correct memory chip during read/write:
  - Data usually stored in pairs of chips.
  - Each chip partially connected to data bus.
  - $\overline{\text{LDS}}$ activates D0 to D7 (odd bytes).
  - $\overline{\text{UDS}}$ activates D8 to D15 (even bytes).

How Data is Stored in Memory

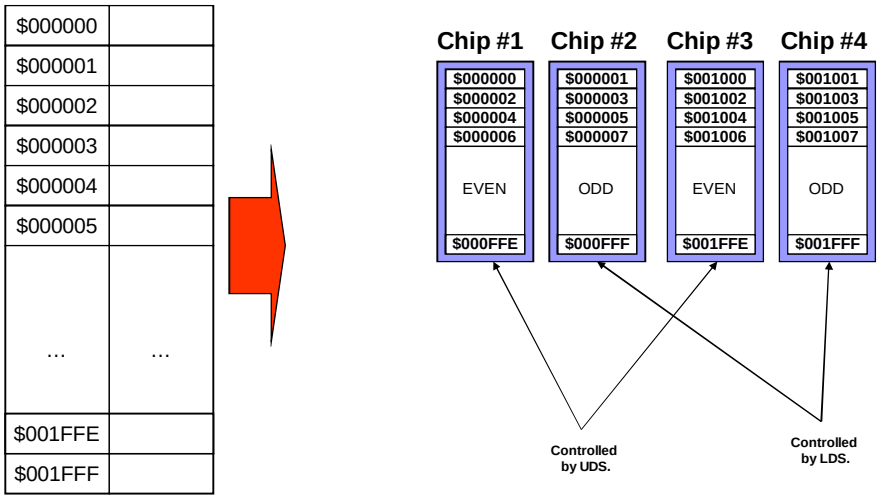


How Data is Stored in Memory

- MOVE.L #$12345678,$1000

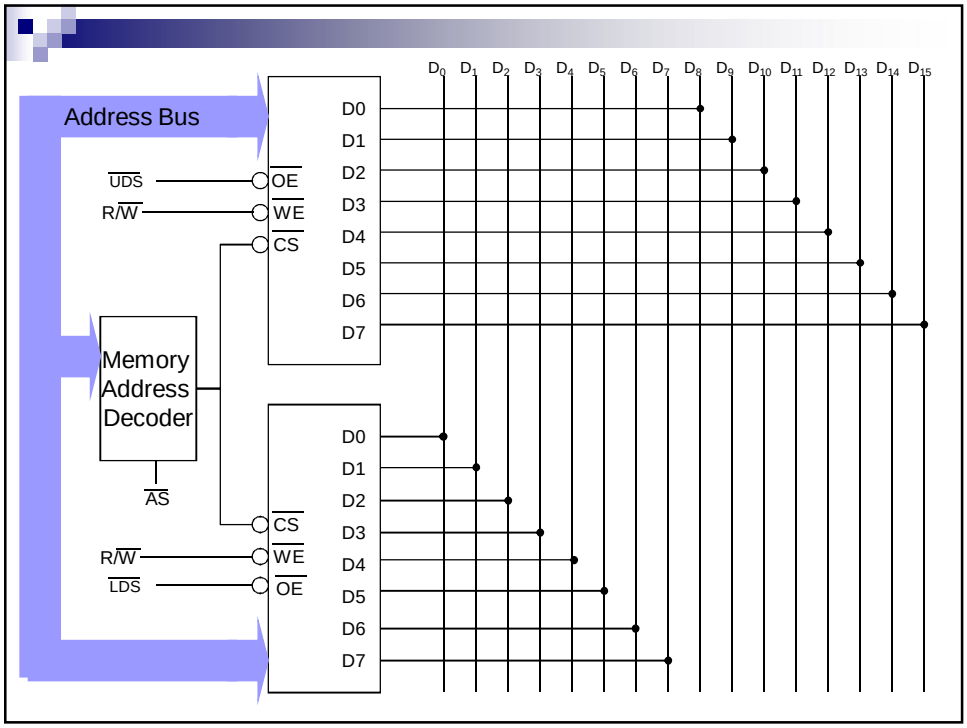


How Data is Stored in Memory



Which one gets selected?

$\overline{UDS}$	$\overline{LDS}$	$\overline{CS}$	What data to access	Chip selected
1	1	0	No valid data	None
0	1	0	D8-D15 (even bytes)	Chip #1
1	0	0	D0-D7 (odd bytes)	Chip #2
0	0	0	D0-D15 (word data)	Both
X	X	1	None ($\overline{CS}$ high)	None



Designing with Available Memories

Types of Memory

- Typically, M68k systems are designed with 2 types of memory:
  - EPROM: Erasable Programmable Read-Only Memory.
  - SRAM: Static Random Access Memory.
- To design a memory system, it's important to know the name and size of chip.

Typical EPROM Chips

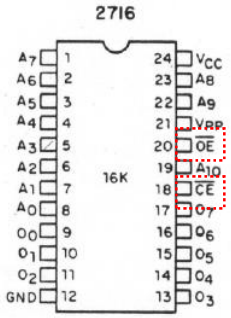
Part	Size	Address Lines	Other Name
2716	2kB	11	2K x 8 ROM
2732	4kB	12	4K x 8 ROM
2764	8kB	13	8K x 8 ROM
27128	16kB	14	16K x 8 ROM
27256	32kB	15	32K x 8 ROM
27512	64kB	16	64K x 8 ROM

Typical SRAM Chips

Part	Size	Address Lines	Other Name
6116	2kB	11	2K x 8 RAM
6264	8kB	13	8K x 8 RAM
62256	32kB	15	32K x 8 RAM

Typical EPROM Chips

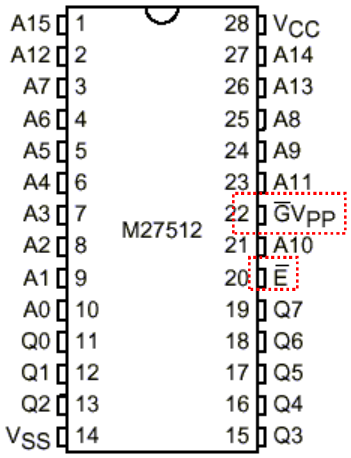
2716 EPROM



PIN NAMES

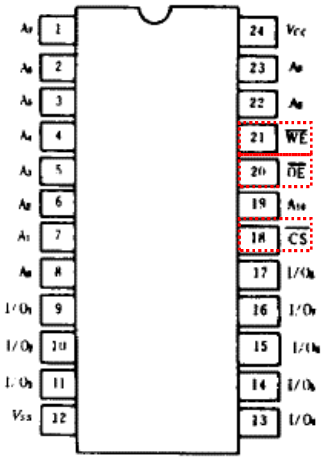
A_0-A_{10}	ADDRESSES
CE/PGM	CHIP ENABLE/PROGRAM
OE	OUTPUT ENABLE
O_0-O_7	OUTPUTS

27512 EPROM

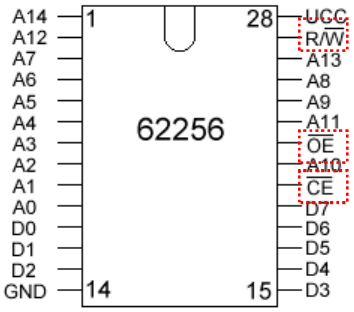


Typical SRAM Chips

62128 SRAM



62256 SRAM

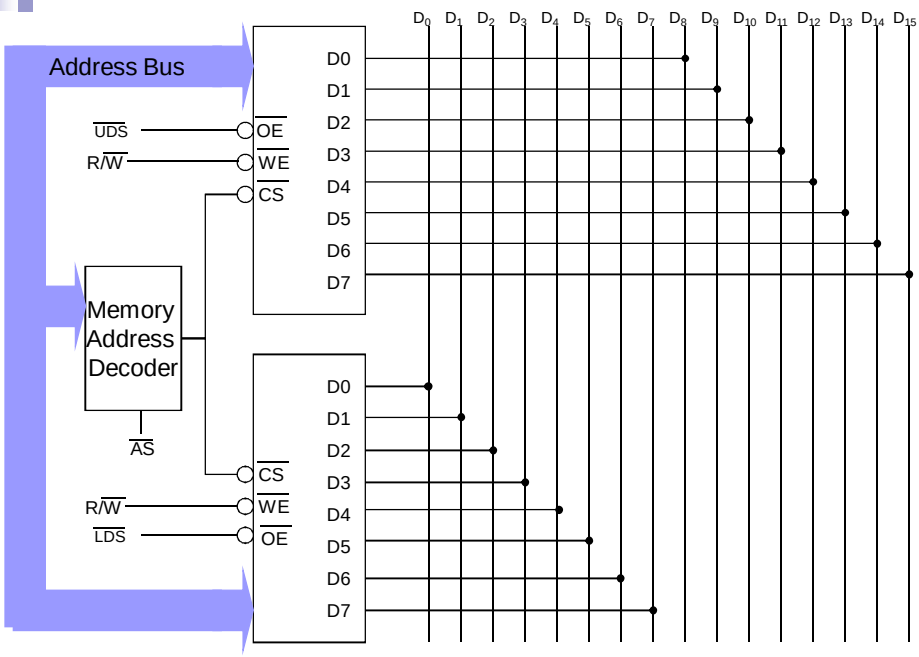


Characteristics

- ROM chips have OE* (output enable) and CS* (chip select).
- RAM chips have E* (enable), CS* (chip select), and WE* (write enable).
- To activate a RAM/ROM chip, both OE*/E* and CS* must be active.

Characteristics

- OE*/E* is connected to UDS*/LDS*.
- CS* is connected to Memory Address Decoder.
- WE* is connected to R/W* pin on M68k.
  - If R/W* = 1, read from memory.
  - If R/W* = 0, write to memory.

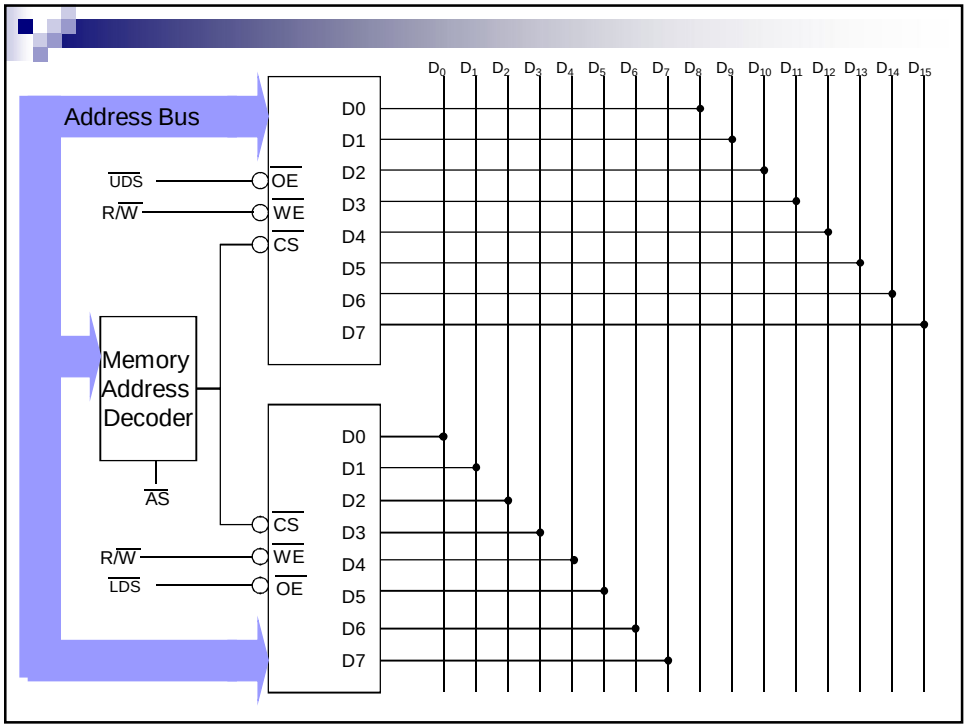


Activating the Correct Chip

- Each memory chip has:
  - Data pins: outputs/receives to/from M68k.
  - Address pins: receives address from M68k.
  - $\overline{OE}$ (Output Enable): Allows data to be sent from chip.
  - $\overline{CS}$ (Chip Select): Enables chip for data transfer.
  - $\overline{WE}$ (Write Enable) (for RAM only): Allows data to be written to chip.

Activating the Correct Chip

- Chip activated only if:
  - $\overline{CS}$ is enabled.
  - $\overline{OE}$ is enabled.
- $\overline{CS}$ enabled when MAD receives unique pattern from Address Bus, $\overline{AS}$ active.
- $\overline{OE}$ enabled when $\overline{UDS}$ or $\overline{LDS}$ active.



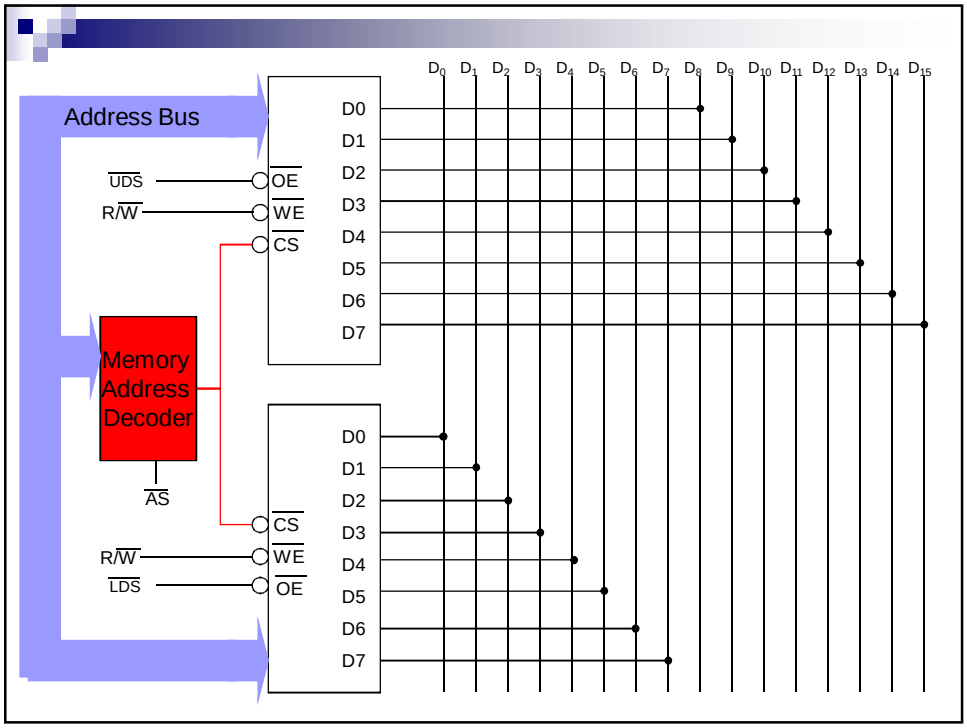
Memory Decoding

Memory Decoding

- Design method to allow M68k to access data in memory.
- Enables/disables certain chips based on data required.
- Done using special circuit – Memory Address Decoder (MAD).

Memory Address Decoder (MAD)

- Special circuit connected to part of address bus.
- Activates specific memory chips based on address pattern in address bus.
- 2 methods to design:
  - Full address decoding (FAD).
  - Partial address decoding (PAD).



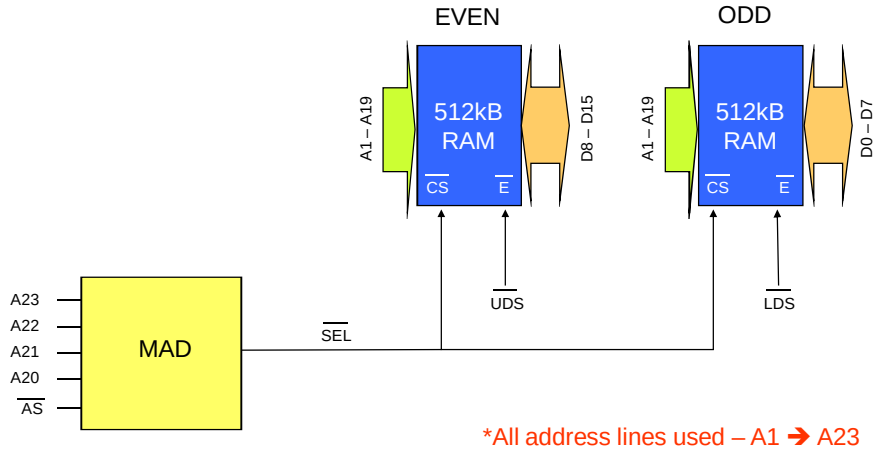
Full Address Decoding

- Uses all available address lines to design decoder:
  - All 23 lines used for addressing/decoding.
- Since all lines used, decoder circuit design is more complex, but easy to expand as extra memory is added.
- Typically used in systems with large memory.

Partial Address Decoding

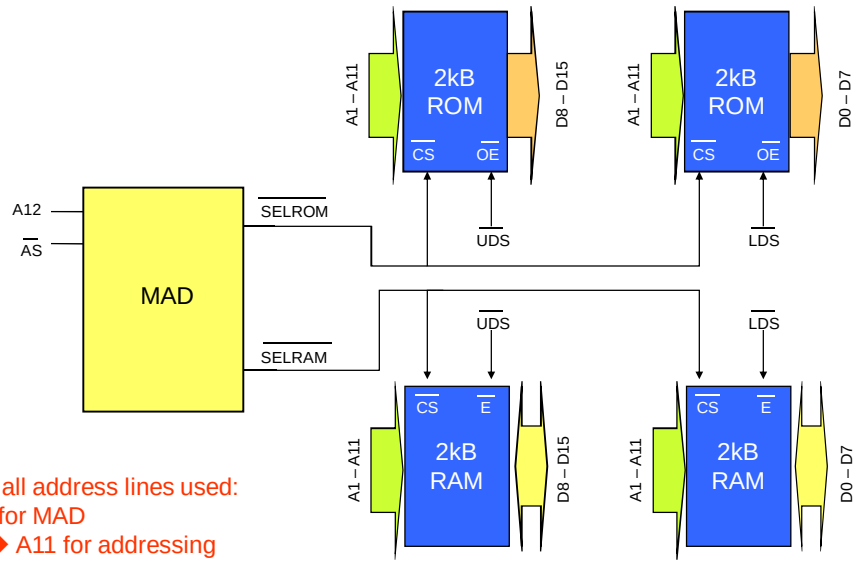
- Uses only uses a few address lines to design decoder:
  - Some lines are not connected to decoder circuit.
- Typically used in systems with smaller memory.
- Decoder circuit design is simple, but hard to expand if more memory needs to be added later.

Full Address Decoding Example



*All address lines used – A1 ➔ A23
A20 ➔ A23 for MAD
A1 ➔ A19 for addressing

Partial Address Decoding Example



*Not all address lines used:
A12 for MAD
A1 ➔ A11 for addressing

Comparison between FAD and PAD

	Full Address Decoder	Partial Address Decoder
Address lines used	Uses all lines for addressing or decoding.	Uses only a part of address lines for addressing or decoding. The rest is ignored.
MAD circuit	More complex circuit, since need to use all address lines.	Less complex circuit, since only a few address lines are used.
When to use	When the M68k system is large and requires a lot of memory.	When M68k system has small memory requirements.
Upgrade	Easy to upgrade, extra memory can be added with little modifications to original decoder.	Difficult to upgrade, requires complete redesign of decoder.

Full Address Decoder Design

Full Address Decoder Design

1. Determine available information.
2. Determine the required number of address lines.
3. Set base address.
4. Determine lower address range.
5. Determine upper address range.
6. Design decoder.
7. Draw memory block diagram.

Example #1

Example #1

- 512kWords (1024 kB) of RAM needs to be interfaced to a 68k-based system, The base address is $400000. Design the decoder circuit.

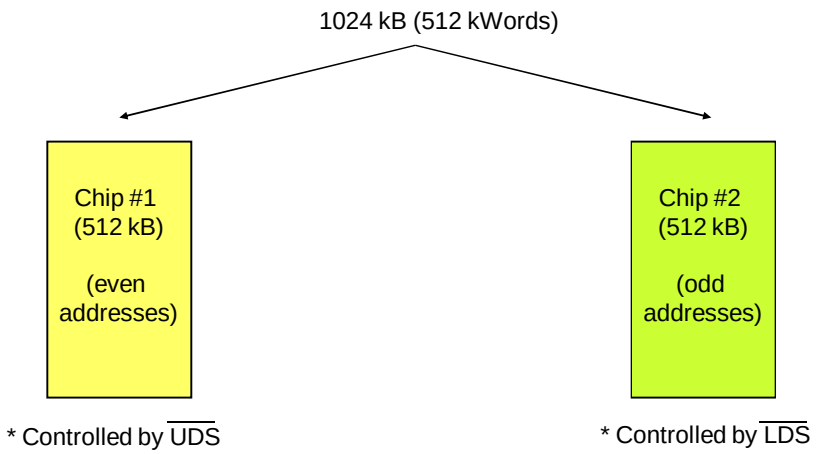
Step 1: Determine Available Information

- Three things must be determined:
  - How much memory to interface.
  - Base address of memory.
  - How many chips need to be used.

Step 1: Determine Available Information

- 1024 kB need to be interfaced:
  - RAM needs to be interfaced.
  - 2 chips used.
  - 512kB for even address ($\overline{UDS}$), 512kB for odd address ($\overline{LDS}$).
- Base address is $400000.

Step 1: Determine Available Information



Step 2: Determine Number of Required Address Lines

- Determines how many address lines need to be used by one chip.
- Use the following formula:

$$x = \frac{\log_{10} y}{\log_{10} 2}$$

y = storage size of one chip (kB)
x = number of reserved lines

Step 2: Determine Number of Required Address Lines

- Each chip contains 512,000 memory locations:
  - Needs 19 address lines.

$$2^x = 512{,}000$$

$$x \log_{10} 2 = \log_{10} 512{,}000$$

$$x = \frac{\log_{10} 512{,}000}{\log_{10} 2} = \frac{5.7093}{0.3010} = 18.97 \approx 19$$

*Always round to higher.

Step 3: Allocate Address Line

- Address lines allocated based on Step 2.
- Start with A1.
- Fill with don't cares (X).

Step 3: Allocate Address Line

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

19 lines allocated

$\overline{\text{UDS}}$/$\overline{\text{LDS}}$
(reserved)

Step 4: Set Base Address

- Set base address using the remaining address lines.

Step 4: Set Base Address

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

4

$\overline{\text{UDS}}$/$\overline{\text{LDS}}$
(reserved)

* Base address is $400000

Step 5: Determine Lower Address Range

- Replace all don't cares and A0 with zeros.
- Should get the same base address as question.

Step 5: Determine Lower Address Range

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4 | Fill with zeros

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4 | 0 | 0 | 0 | 0 | 0

Lower range: $400000

Step 6: Determine Upper Address Range

- Replace all don't cares and A0 with ones.
- This determines the upper limit of the memory chip address.

Step 6: Determine Upper Address Range

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4 — Fill with ones

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4 | F | F | F | F | F

Upper range: $4FFFFF

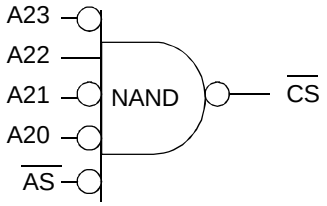
Step 7: Design Decoder using Remaining Addresses

- The decoder must be designed so that only the specific bit combinations in the base address can generate a zero on $\overline{CS}$ (output).
- Typically uses NAND gate.
- $\overline{AS}$ must be included together in decoder.

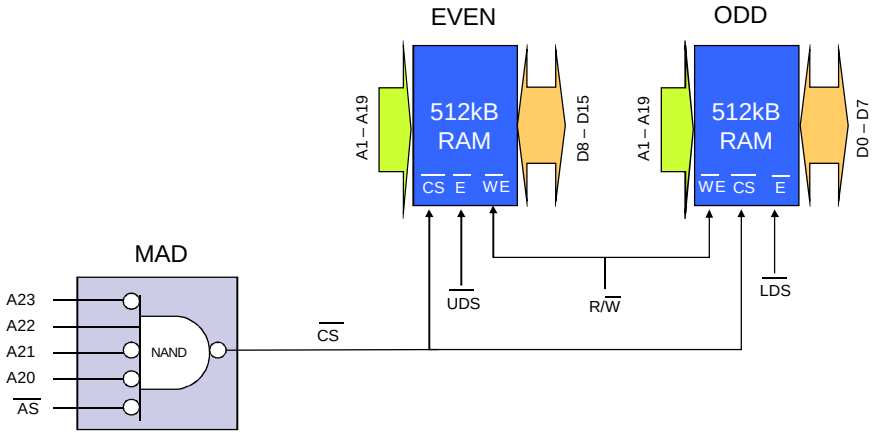
Step 7: Design Decoder using Remaining Addresses

A23	A22	A21	A20
0	1	0	0

4



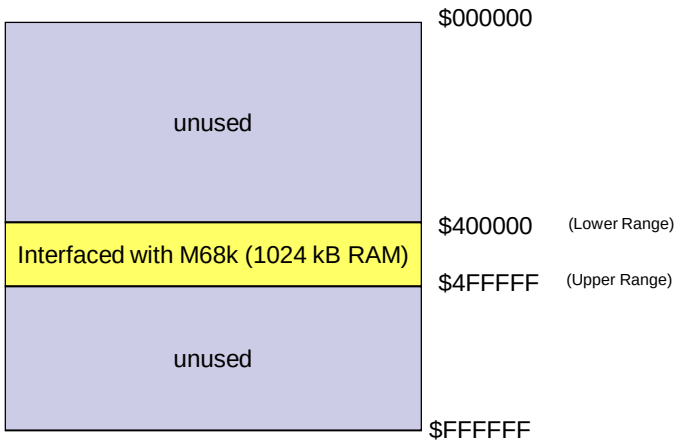
Actual Implementation



Step 8: Draw Memory Block Diagram

- Memory block diagram shows assignment of memory addresses.
- Begins at $000000, ends at $FFFFFF.
- Mark the locations where memory has been interfaced.

Memory Block Diagram



Example #2

Example #2

- 8k Words (16 kB) of ROM needs to be interfaced to a 68k-based system, so that the base ROM address is at $AE0000. Determine the address range and design the decoder circuit.

Step 1: Determine Available Information

- 8 kWords (16kB) need to be interfaced:
  - 2 chips need to be used.
  - 8kB for even address, 8kB for odd address.
- Base address is $AE0000.

Step 2: Determine Number of Required Address Lines

- Each chip contains 8,000 (8kB) memory locations:
  - Needs 13 address lines.

$$2^x = 8000$$

$$x \log_{10} 2 = \log_{10} 8000$$

$$x = \frac{\log_{10} 8000}{\log_{10} 2} = \frac{3.9031}{0.3010} = 12.97 \approx 13$$

*Always round to higher.

Step 3: Allocate Address Line

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
										X	X	X	X	X	X	X	X	X	X	X	X	X	

13 lines allocated

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 4: Set Base Address

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	

A

E

Unused, fill with zeros

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 5: Determine Lower Address Range

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A				E						Fill with zeros													

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A				E				0				0				0				0			

Lower range: $AE0000

Step 6: Determine Upper Address Range

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
A				E						Fill with ones													

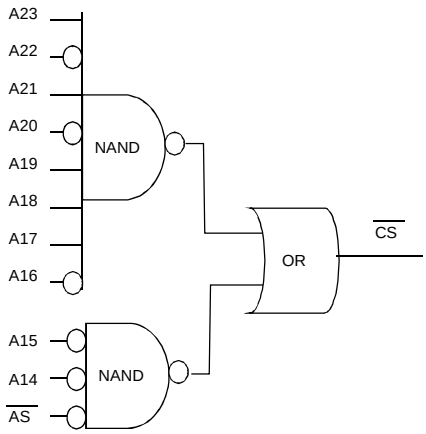
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
A				E				3				F				F				F			

Upper range: $AE3FFF

Step 7: Design Decoder using Remaining Addresses

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
1	0	1	0	1	1	1	0	0	0

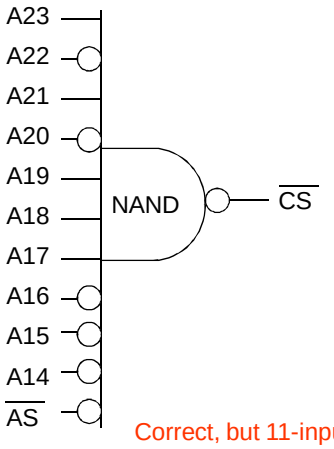
A E



Why can't we do it like this?

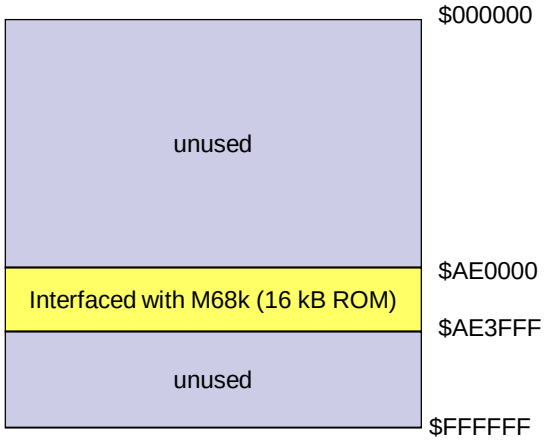
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
1	0	1	0	1	1	1	0	0	0

A E

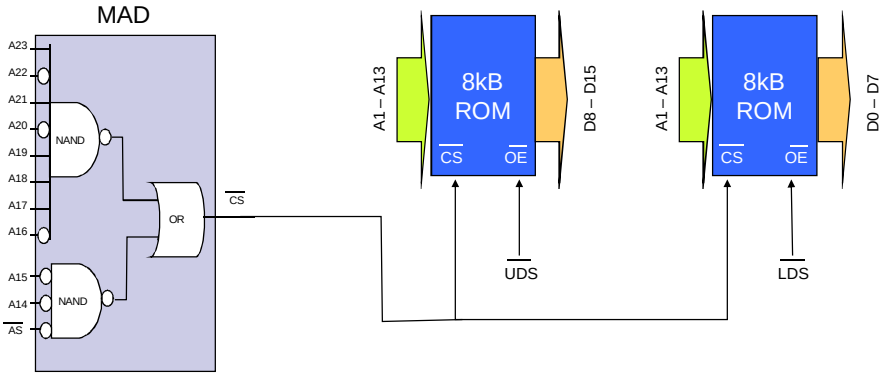


Correct, but 11-input NAND gate doesn't exist (max = 8).

Memory Block Diagram



Actual Implementation



Example #3

Example #3

- Show how 32kB of EPROM and 32kB of SRAM can be interfaced to the M68k to implement a system containing 64kB of ROM commencing at location $C00000 and 32kW of RAM commencing at location $300000.

Step 1: Determine Available Information

- 64 kB of ROM & 32kW of RAM need to be interfaced:
  - 4 chips need to be used.
  - 32kB for even ROM, 32kB for odd ROM.
  - 32kB for even RAM, 32kB for odd ROM.
- ROM base address is $C00000.
- SRAM base address is $300000.

Step 2(a): Determine Number of Required Address Lines for ROM

- Each ROM chip contains 32,000 memory locations:
  - Needs 15 address lines.

$$2^x = 32{,}000$$

$$x \log_{10} 2 = \log_{10} 32{,}000$$

$$x = \frac{\log_{10} 32{,}000}{\log_{10} 2} = \frac{4.5051}{0.3010} = 14.97 \approx 15$$

*Always round to higher.

Step 2(b): Determine Number of Required Address Lines for RAM

- Each RAM chip contains 32,000 memory locations:
  - Needs 15 address lines.

$$2^x = 32{,}000$$

$$x \log_{10} 2 = \log_{10} 32{,}000$$

$$x = \frac{\log_{10} 32{,}000}{\log_{10} 2} = \frac{4.5051}{0.3010} = 14.97 \approx 15$$

*Always round to higher.

Step 3: Allocate Address Line

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
								X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

15 lines allocated

$\overline{UDS}$/$\overline{LDS}$ (reserved)

RAM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
								X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

15 lines allocated

$\overline{UDS}$/$\overline{LDS}$ (reserved)

Step 4: Set Base Address

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
C				0				15 lines allocated															UDS/LDS (reserved)

RAM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
3				0				15 lines allocated															UDS/LDS (reserved)

Step 5(a): Determine Lower Address Range (ROM)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C				0				Fill with zeros															

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C				0				0				0				0				0			

ROM Lower range: $C00000

Step 5(b): Determine Lower Address Range (RAM)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3				0				Fill with zeros															

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3				0				0				0				0				0			

RAM Lower range: $300000

Step 6(a): Determine Upper Address Range (ROM)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C				0				Fill with ones															

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C				0				F				F				F				F			

ROM upper range: $C0FFFF

Step 6(b): Determine Upper Address Range (RAM)

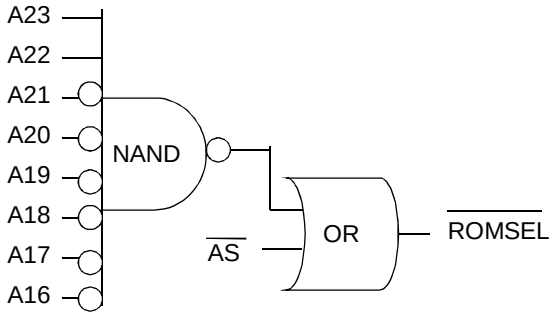
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3				0				Fill with ones															

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3				0				F				F				F				F			

RAM upper range: $30FFFF

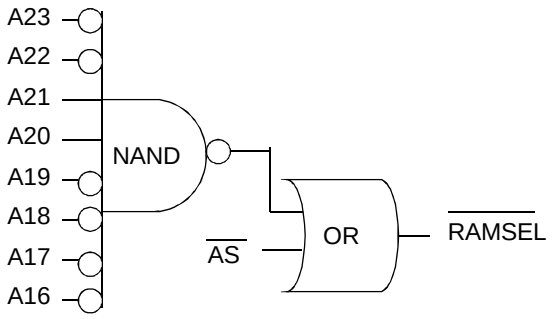
Step 7(a): Design Decoder using Remaining Addresses (ROM)

A23	A22	A21	A20	A19	A18	A17	A16
1	1	0	0	0	0	0	0
C				0			

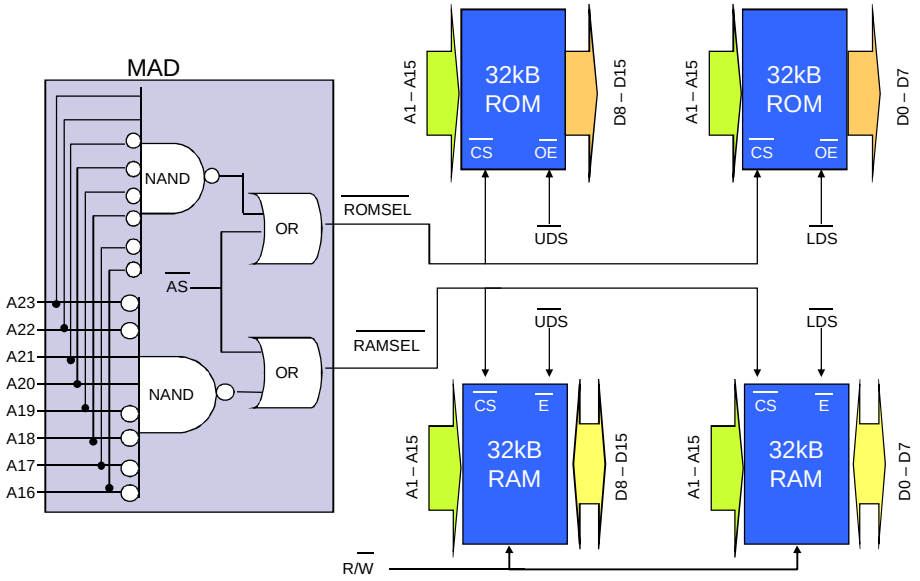


Step 7(b): Design Decoder using Remaining Addresses (RAM)

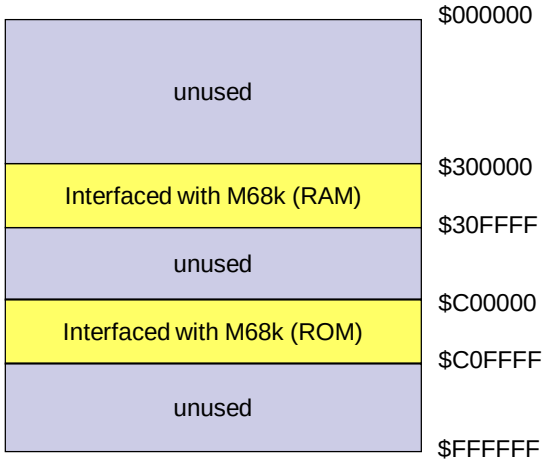
A23	A22	A21	A20	A19	A18	A17	A16
0	0	1	1	0	0	0	0
3				0			



Actual Implementation



Memory Block Diagram



Example #4

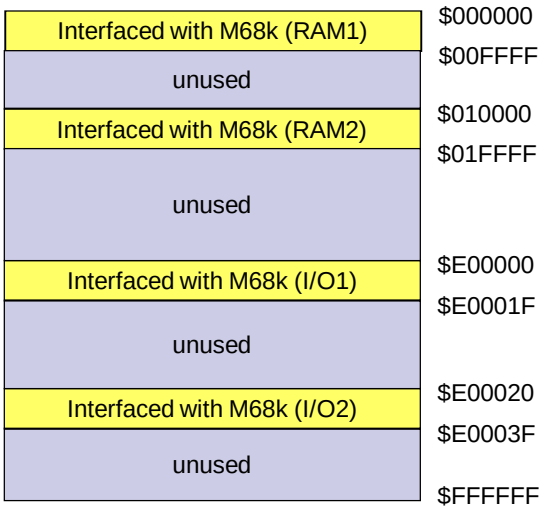
Example #4

- Design an address decoding network to satisfy the following memory map:
  - RAM1: $000000 - $00FFFF.
  - RAM2: $010000 - $01FFFF.
  - I/O1: $E00000 - $E0003F.
  - I/O2: $E00040 - $E0007F.

Determine Available Information

- 2 RAM and 2 I/O locations need to be interfaced.
- Memory address range already given:
  - Start of base address not given.
  - Start of required address lines not given.
  - Have to determine by examining memory range.

Memory Block Diagram



RAM1 Address Range

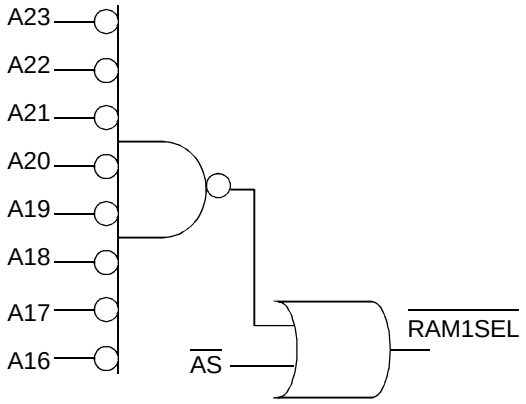
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				F				F				F				F			

* Required address lines: A1 ➔ A15, Decoder address lines: A16 ➔ A23

RAM1 Decoder

A23	A22	A21	A20	A19	A18	A17	A16
0	0	0	0	0	0	0	0
0				0			



RAM2 Address Range

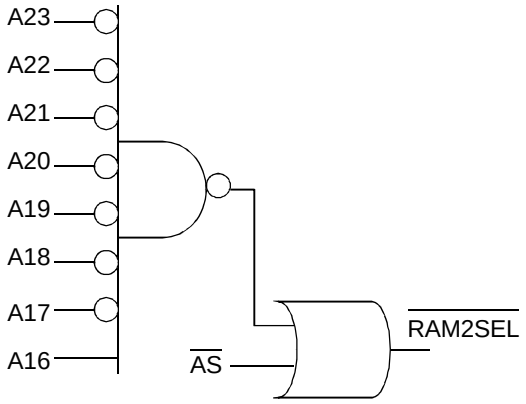
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				1				0				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				1				F				F				F				F			

* Required address lines: A1 ➔ A15, Decoder address lines: A16 ➔ A23

RAM2 Decoder

A23	A22	A21	A20	A19	A18	A17	A16
0	0	0	0	0	0	0	1
0				1			



I/O1 Address Range

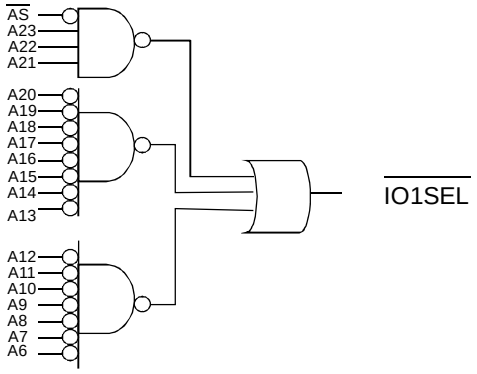
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E				0				0				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
E				0				0				0				3				F			

* Required address lines: A1 ➔ A5, Decoder address lines: A6 ➔ A23

I/O1 Decoder

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E				0				0				0					



I/O2 Address Range

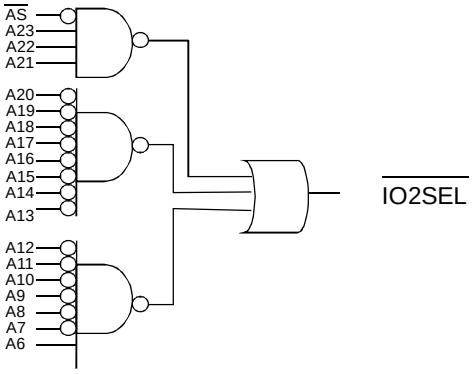
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
E				0				0				0				2				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
E				0				0				0				7				F			

* Required address lines: A1 ➔ A5, Decoder address lines: A6 ➔ A23

I/O2 Decoder

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
E				0				0				0					



Example #5

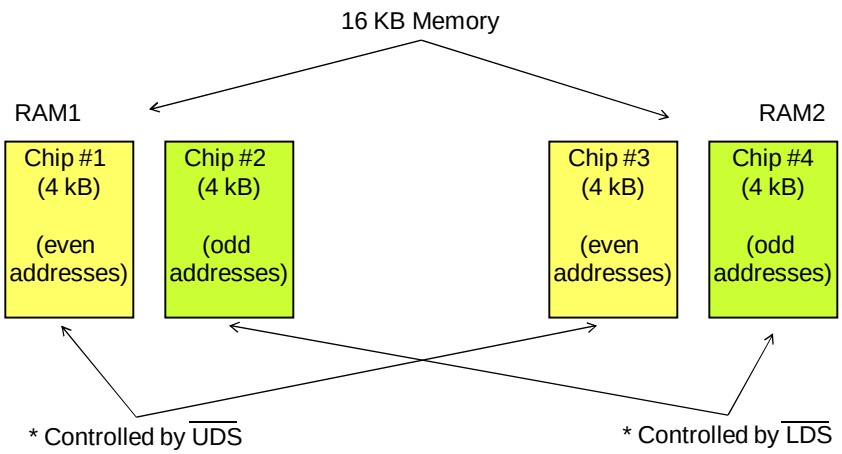
Example #5

- Design a 16kB M68k memory system using four 32K x 8 chips. The base address is $C000.

Step 1: Determine Available Information

- Four 32K x 8 chips need to be interfaced.
- Each chip is 4kB in size.
- Base address is $C000.

Step 1: Determine Available Information



Step 2: Determine Number of Required Address Lines

- Each chip contains 4,000 memory locations:
  - Needs 19 address lines.

$$2^x = 4{,}000$$

$$x \log_{10} 2 = \log_{10} 4{,}000$$

$$x = \frac{\log_{10} 4{,}000}{\log_{10} 2} = \frac{3.6021}{0.3010} = 11.96 \approx 12$$

*Always round to higher.

Step 3: Allocate Address Line

RAM1

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
											X	X	X	X	X	X	X	X	X	X	X	X	

RAM2

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
											X	X	X	X	X	X	X	X	X	X	X	X	

Step 4: Set Base Address

RAM1

A23	A22	A21	A20	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	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	

RAM2

A23	A22	A21	A20	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	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	

RAM1 pair begins at $00C000 and ends at $00DFFF. RAM2 pair should begin at the next memory location, which is $00E000, and ends at $00FFFF.

Step 5: Determine Lower Address Range

RAM1

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RAM2

A23	A22	A21	A20	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	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Step 6: Determine Upper Address Range

RAM1

A23	A22	A21	A20	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	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

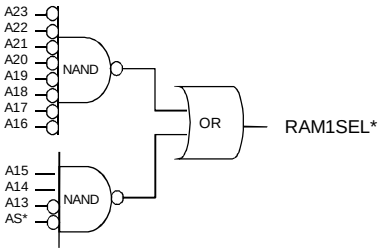
RAM2

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Step 7: Design Decoder using Remaining Addresses

RAM1

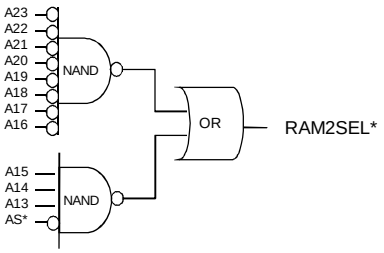
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
0	0	0	0	0	0	0	0	1	1	0



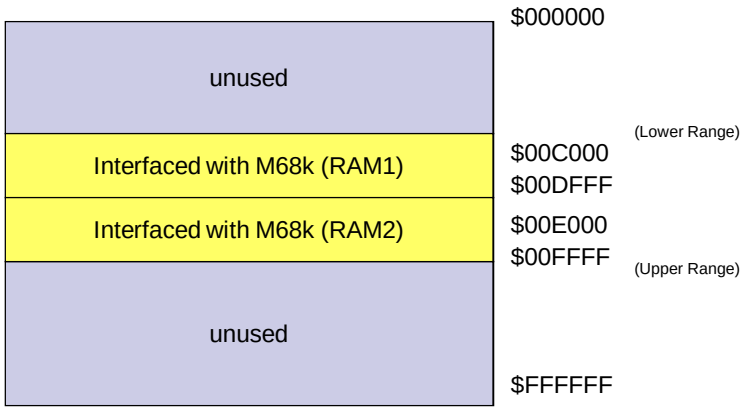
Step 7: Design Decoder using Remaining Addresses

RAM2

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
0	0	0	0	0	0	0	0	1	1	1



Memory Block Diagram



Designing A Partial Address Decoder

Design Steps

1. Determine available information.
2. Determine the required number of address lines.
3. Allocate address lines.
4. Set base address.
5. Determine lower address range.
6. Determine upper address range.
7. Find unique pattern in the address lines.
8. Design decoder.
9. Draw memory block diagram.

Example #1

Example #1

- M68k needs to be interfaced with 4kW of memory (2kW EPROM, 2kW RAM). The ROM base address is $1000 and the RAM base address is $3000. Design the Partial Address Decoder.

Step 1: Determine Available Information

- 4kW of memory needs to be interfaced:
  - 2kW ROM = 2 x 2kB EPROM
  - 2kW RAM = 2 x 2kB RAM
  - 4 chips need to be interfaced.
- Starting address $1000 (ROM), $3000 (RAM).

Step 2: Determine Number of Required Address Lines (RAM & ROM)

- Each chip contains 2,000 memory locations:
  - Needs 11 address lines.

$$2^x = 2{,}000$$

$$x \log_{10} 2 = \log_{10} 2{,}000$$

$$x = \frac{\log_{10} 2{,}000}{\log_{10} 2} = \frac{3.3010}{0.3010} = 10.97 \approx 11$$

*Always round to higher.

Step 3: Allocate Address Line

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
												X	X	X	X	X	X	X	X	X	X	X	

11 lines allocated

$\overline{UDS}/\overline{LDS}$ (reserved)

RAM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
												X	X	X	X	X	X	X	X	X	X	X	

11 lines allocated

$\overline{UDS}/\overline{LDS}$ (reserved)

Step 4: Set Base Address

ROM

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	
0				0				1															

$\overline{UDS}/\overline{LDS}$
(reserved)

RAM

A23	A22	A21	A20	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	1	1	X	X	X	X	X	X	X	X	X	X	X	
0				0				3															

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 5: Determine Lower Address Range

ROM

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				1				0				0				0			

RAM

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				3				0				0				0			

Step 6: Determine Upper Address Range

ROM

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				1				F				F				F			

RAM

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				3				F				F				F			

Step 7: Find Unique Pattern

- To find the unique pattern, cancel out any similar address lines for all the memory ranges being interfaced.
- The remaining pattern should be a combination of address lines that uniquely identifies each memory range.

Step 7: Find Unique Pattern

ROM

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	
C	C	C	C	C	C	C	C	C	C		C												

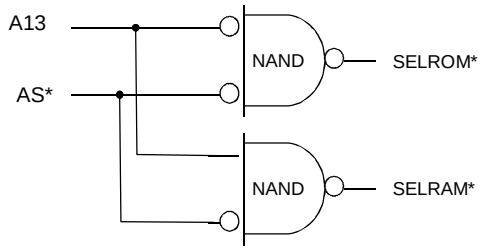
RAM

A23	A22	A21	A20	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	1	1	X	X	X	X	X	X	X	X	X	X	X	
C	C	C	C	C	C	C	C	C	C		C												

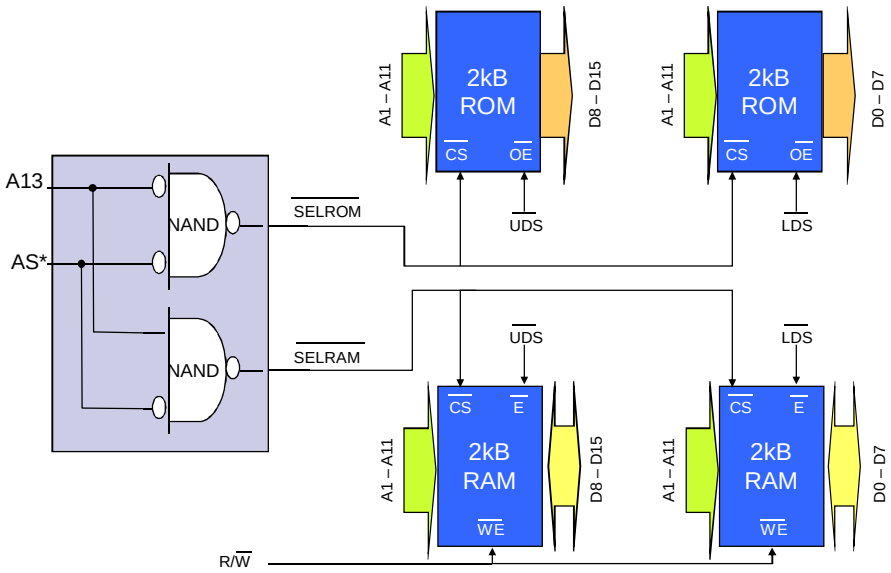
Step 8: Design Decoder Circuit

A13	$\overline{AS}$	$\overline{SELROM}$	$\overline{SELRAM}$
0	0	0	1
0	1	1	1
1	0	1	0
1	1	1	1

To select ROM, A13 = 0, and $\overline{AS}$ = 0.
To select RAM, A13 = 1, and $\overline{AS}$ = 0



Actual Implementation



Example #2

Example #2

- 64kB RAM and 32kB ROM need to be interfaced with a M68k-system. The ROM starting address is $400000. The RAM starting address is $FF0000. Design the decoder using partial addressing method.

Step 1: Determine Available Information

- 96kB of memory needs to be interfaced:
  - 32kB ROM = 2 x 16kB EPROM
  - 64kB RAM = 2 x 32kB RAM
  - 4 chips need to be interfaced.
- Starting address $400000 (ROM), $FF0000 (RAM).

Step 2a: Determine Number of Required Address Lines (RAM)

- Each chip contains 32,000 memory locations:
  - Needs 15 address lines.

$$2^x = 32{,}000$$

$$x \log_{10} 2 = \log_{10} 32{,}000$$

$$x = \frac{\log_{10} 32{,}000}{\log_{10} 2} = \frac{4.5051}{0.3010} = 14.97 \approx 15$$

*Always round to higher.

Step 2b: Determine Number of Required Address Lines (ROM)

- Each chip contains 16,000 memory locations:
  - Needs 14 address lines.

$$2^x = 16{,}000$$

$$x \log_{10} 2 = \log_{10} 16{,}000$$

$$x = \frac{\log_{10} 16{,}000}{\log_{10} 2} = \frac{4.2041}{0.3010} = 13.96 \approx 14$$

*Always round to higher.

Step 3: Allocate Address Line

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
									X	X	X	X	X	X	X	X	X	X	X	X	X	X	

14 lines allocated

$\overline{UDS}/\overline{LDS}$ (reserved)

RAM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
								X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

15 lines allocated

$\overline{UDS}/\overline{LDS}$ (reserved)

Step 4: Set Base Address

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

4 0

$\overline{UDS}/\overline{LDS}$ (reserved)

RAM

A23	A22	A21	A20	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	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

F F

$\overline{UDS}/\overline{LDS}$ (reserved)

Step 5: Determine Lower Address Range

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4				0				0				0				0				0			

RAM

A23	A22	A21	A20	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F				F				0				0				0				0			

Step 6: Determine Upper Address Range

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4				0				7				F				F				F			

RAM

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F				F				F				F				F				F			

Step 7: Find Unique Pattern

ROM

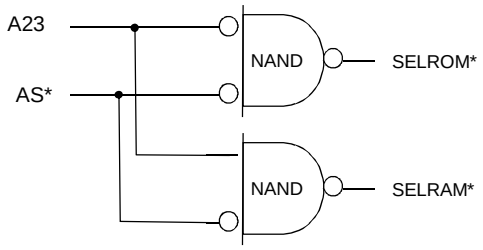
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4				0				7				F				F				F			

RAM

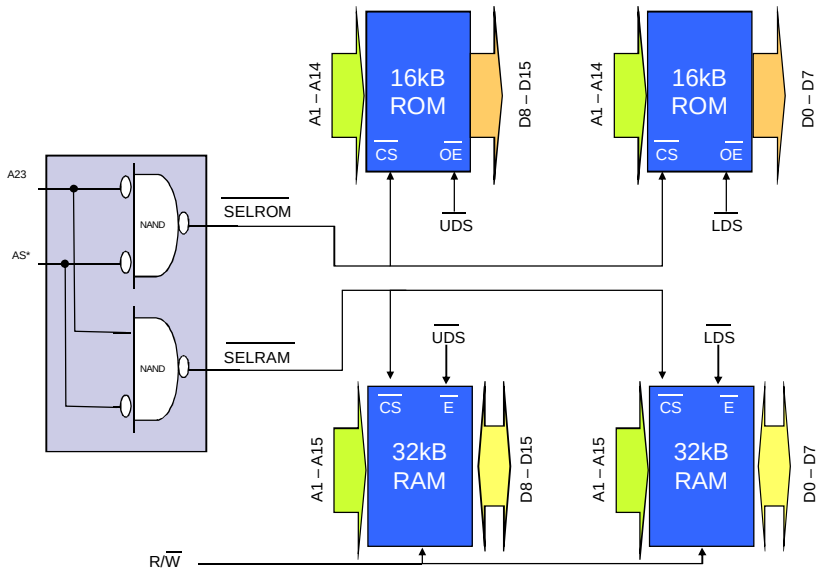
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F				F				F				F				F				F			

Step 8: Design Decoder Circuit

To select ROM, A23 = 0, and $\overline{AS}$ = 0.
To select RAM, A23 = 1, and $\overline{AS}$ = 0



Actual Implementation



Example #3

Example #3

- A small system need to interface memory to a 68k-based system. The address ranges are:
  - ROM: $0000 - $07FF
  - RAM: $2000 – $2FFF
  - I/O: $A000 - $A03F
- Design the memory address decoder using Partial Addressing method.

Step 1: Determine Available Information

- 3 types of memory locations need to be decoded.
- Memory size not given.
- Address range given:
  - ROM: $0000 - $07FF
  - RAM: $2000 – $2FFF
  - I/O: $A000 - $A03F

Step 2a: Determine Number of Required Address Lines (ROM)

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Lower Range	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0				0				0				0			
Upper Range	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	0				7				F				F			
To Decoder ➔	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X

Step 2b: Determine Number of Required Address Lines (RAM)

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Lower Range	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	2				0				0				0			
Upper Range	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	2				F				F				F			
To Decoder ➔	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X

Step 2c: Determine Number of Required Address Lines (I/O)

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Lower Range	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	A				0				0				0			
Upper Range	1	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1
	A				0				3				F			
To Decoder ➔	1	0	1	0	0	0	0	0	0	0	X	X	X	X	X	X

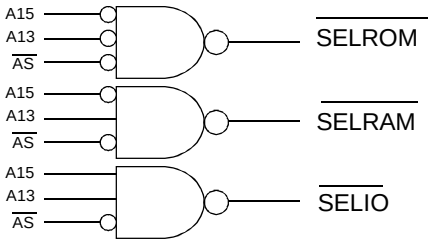
Step 7: Find Unique Pattern

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ROM	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X
RAM	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
I/O	1	0	1	0	0	0	0	0	0	0	X	X	X	X	X	X

A15	A13	$\overline{AS}$	
0	0	0	$\overline{SELROM}$
0	1	0	$\overline{SELRAM}$
1	1	0	$\overline{SELIO}$
X	X	1	All Disabled

Step 8: Design Decoder Circuit

A15	A13	$\overline{AS}$	Output
0	0	0	$\overline{SELROM}$ = 0 (activated)
0	1	0	$\overline{SELRAM}$ = 0 (activated)
1	1	0	$\overline{SELIO}$ = 0 (activated)
X	X	1	All Disabled



Example #4

Example #4

- A M68k system needs to be built with the following specifications:
  - EPROM: 4kB needed, start address $1000.
  - SRAM: 2kB needed, start address $2000.
  - I/O, start address $3000.
- Design the decoder using partial address decoding.

Step 1: Determine Available Information

- Memory that needs to be interfaced:
  - 4 kB ROM = 2 x 2kB ROM.
  - 2 kB RAM = 2 x 1kB RAM
  - I/O = 5 lines automatically reserved.
- Starting address $1000 (ROM), $2000 (RAM), $3000 (I/O).

Step 2a: Determine Number of Required Address Lines (ROM)

- Each chip contains 2,000 memory locations:
  - Needs 11 address lines.

$$2^x = 2{,}000$$

$$x \log_{10} 2 = \log_{10} 2{,}000$$

$$x = \frac{\log_{10} 2{,}000}{\log_{10} 2} = \frac{3.3010}{0.3010} = 10.97 \approx 11$$

*Always round to higher.

Step 2b: Determine Number of Required Address Lines (RAM)

- Each chip contains 1,000 memory locations:
  - Needs 10 address lines.

$$2^x = 1{,}000$$

$$x \log_{10} 2 = \log_{10} 1{,}000$$

$$x = \frac{\log_{10} 1{,}000}{\log_{10} 2} = \frac{3}{0.3010} = 9.97 \approx 10$$

*Always round to higher.

Step 2c: Determine Number of Required Address Lines (I/O)

- I/O automatically requires 5 address lines.
- Need to reserve A1 to A5.

Step 3a: Allocate Address Lines (ROM)

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
												X	X	X	X	X	X	X	X	X	X	X	

11 lines allocated

$\overline{UDS}$/$\overline{LDS}$
(reserved)

Step 3b: Allocate Address Lines (RAM)

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
													X	X	X	X	X	X	X	X	X	X	

10 lines allocated

$\overline{UDS}$/$\overline{LDS}$ (reserved)

Step 3c: Allocate Address Lines (I/O)

ROM

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
																		X	X	X	X	X	

5 lines allocated

Step 4a: Set Base Address (ROM)

ROM

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	
0				0				1															

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 4b: Set Base Address (RAM)

ROM

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X	X	X	
0				0				2															

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 4c: Set Base Address (I/O)

ROM

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	X	X	X	X	X	
0				0				3															

$\overline{UDS}/\overline{LDS}$
(reserved)

Step 5a: Determine Lower Address Range (ROM)

ROM

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				1				0				0				0			

Step 5b: Determine Lower Address Range (RAM)

RAM

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				2				0				0				0			

Step 5c: Determine Lower Address Range (I/O)

I/O

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				3				0				0				0			

Step 6a: Determine Upper Address Range (ROM)

ROM

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				1				F				F				F			

Step 6b: Determine Upper Address Range (RAM)

RAM

A23	A22	A21	A20	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	1	0	0	1	1	1	1	1	1	1	1	1	1	1
0				0				2				7				F				F			

Step 6c: Determine Upper Address Range (I/O)

I/O

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	1	1	1	1	1	1
0				0				3				0				3				F			

Step 7: Find Unique Pattern

ROM

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	

RAM

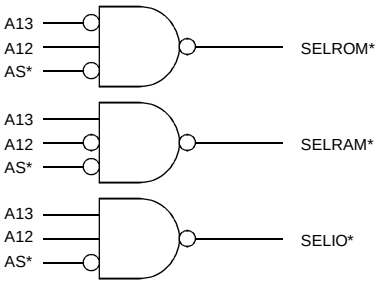
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	

I/O

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	X	X	X	X	X	

Step 8: Design Decoder Circuit

	A13	A12	AS*
ROM	0	1	0
RAM	1	0	0
I/O	1	1	0
	X	X	1



Example #5

Example #5

- Use PAD to design the decoder for these memory ranges:
  - RAM1: $108000 to $1087FF
  - RAM2: $108800 to $108FFF
  - RAM3: $109000 to $1097FF
  - RAM4: $109800 to $109FFF
  - RAM5: $10A000 to $10A7FF
  - RAM6: $10A800 to $108FFF
  - RAM7: $10B000 to $1087FF
  - RAM8: $10B800 to $108FFF

RAM1: 108000 to 1087FF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1				0				8				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1				0				8				7				F				F			

RAM2: 108800 to 108FFF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1				0				8				8				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1				0				8				F				F				F			

RAM3: 109000 to 1097FF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1				0				9				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
1				0				9				7				F				F			

RAM4: 109800 to 109FFF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
1				0				9				8				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1				0				9				F				F				F			

RAM5: 10A000 to 10A7FF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1				0				A				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
1				0				A				7				F				F			

RAM6: 10A800 to 10AFFF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
1				0				A				8				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1				0				A				F				F				F			

RAM7: 10B000 to 10B7FF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1				0				B				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1				0				B				7				F				F			

RAM8: 10B800 to 10BFFF

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1				0				B				8				0				0			

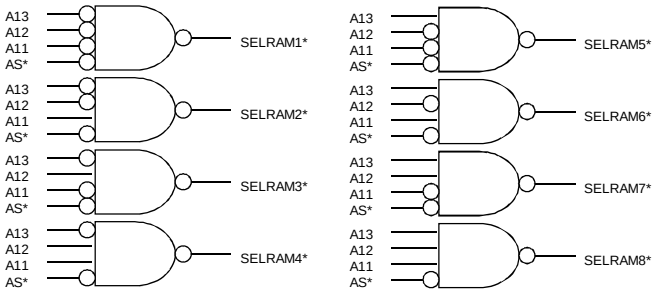
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1				0				B				F				F				F			

Step 7: Find Unique Pattern

0	0	0	1	0	0	0	0	1	0	0	0	0	RAM1
0	0	0	1	0	0	0	0	1	0	0	0	1	RAM2
0	0	0	1	0	0	0	0	1	0	0	1	0	RAM3
0	0	0	1	0	0	0	0	1	0	0	1	1	RAM4
0	0	0	1	0	0	0	0	1	0	1	0	0	RAM5
0	0	0	1	0	0	0	0	1	0	1	0	1	RAM6
0	0	0	1	0	0	0	0	1	0	1	1	0	RAM7
0	0	0	1	0	0	0	0	1	0	1	1	1	RAM8

Step 8: Design Decoder Circuit

0	0	0	RAM1
0	0	1	RAM2
0	1	0	RAM3
0	1	1	RAM4
1	0	0	RAM5
1	0	1	RAM6
1	1	0	RAM7
1	1	1	RAM8



Example #6

Example #6

- Implement a PAD for the following memory map:
  - RAM: $400400 - $4004FF.
  - I/O1: $400500 - $40053F.
  - I/O2: $400540 - $40057F.

Step 1: Determine Available Information

- 3 types of memory locations need to be decoded.
- Memory size not given.
- Address range given:
  - RAM: $400400 - $4004FF.
  - I/O1: $400500 - $40053F.
  - I/O2: $400540 - $40057F.

Step 2a: Determine Number of Required Address Lines (RAM)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4				0				0				4				0				0			

A23	A22	A21	A20	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	1	0	0	1	1	1	1	1	1	1	1
4				0				0				4				F				F			

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

Step 2b: Determine Number of Required Address Lines (I/O1)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
4				0				0				5				0				0			

A23	A22	A21	A20	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	1	0	1	0	0	1	1	1	1	1	1
4				0				0				5				3				F			

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

Step 2b: Determine Number of Required Address Lines (I/O2)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
4				0				0				5				4				0			

A23	A22	A21	A20	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	1	0	1	0	1	1	1	1	1	1	1
4				0				0				5				7				F			

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Step 7: Find Unique Pattern

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Try to cancel out, there are not unique patterns!

Step 7: Find Unique Pattern

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Try to cancel out, there are not unique patterns!

Step 7: Find Unique Pattern

- To design a decoder for these types of memory maps, the decoder has to be designed in two stages.
- The first stage is to design the decoder for RAM, I/O1 and I/O2.
- The second stage is to design the decoder again for I/O1 and I/O2.

Step 7: Find Unique Pattern

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

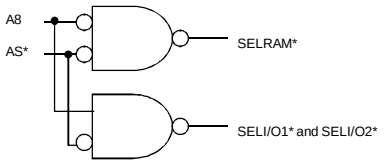
Try to cancel out, there are not unique patterns!

Step 8: Design Decoder Circuit (Stage 1)

A8	A7	A6	A5	A4	A3	A2	A1	A0
0	X	X	X	X	X	X	X	X

A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	X	X	X	X	X	X

A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	X	X	X	X	X	X



Step 8: Design Decoder Circuit (Stage 2)

DONE

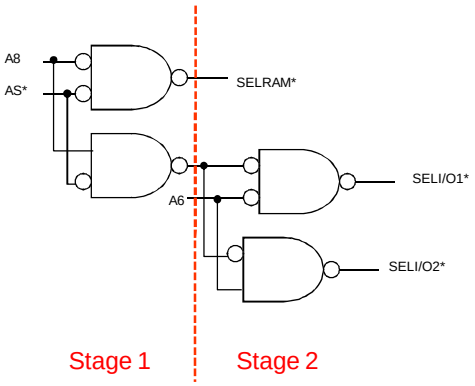
A8	A7	A6	A5	A4	A3	A2	A1	A0
0	X	^	^	^	^	X	X	X

A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	X	X	X	X	X	X

C

A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	X	X	X	X	X	X

C



Designing using Decoder

Designing with Decoders

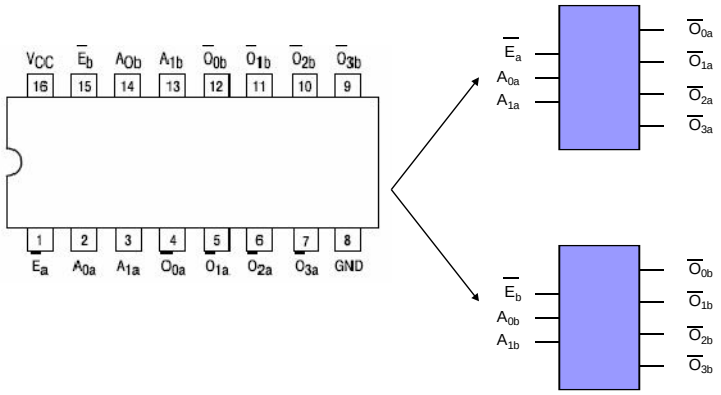
- Other than logic gates, decoders can also be used to create a memory address decoder (MAD).
- Two types of decoders commonly used in M68k systems are:
  - 74LS138 3 to 8 decoder.
  - 74LS139 dual 2 to 4 decoder.

74LS139 Dual 2-4 Line Decoder

- Motorola active low 2-4 decoder.
- 2 x decoders in one IC.
- 16 pins total:
  - 2 x (2 inputs, 4 outputs).
  - Vcc (±5V) and GND.
  - 2 x Enable pins.

Adobe Acrobat 7.0 Document

74LS139 Dual 2-4 Line Decoder



74LS139 Truth Table

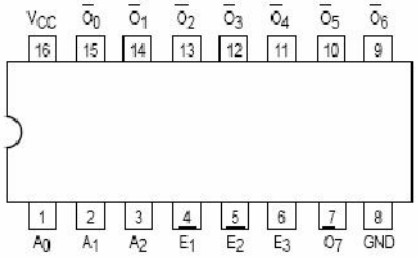
E*	I_1	I_0		O_3^*	O_2^*	O_1^*	O_0^*
1	X	X		1	1	1	1
0	0	0		1	1	1	0
0	0	1		1	1	0	1
0	1	0		1	0	1	1
0	1	1		0	1	1	1

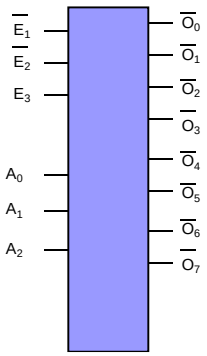
74LS138 3-8 Line Decoder

- Motorola 3-8 active low decoder.
- 1 x decoder in one IC.
- 16 pins total:
  - 3 inputs, 8 outputs (active low).
  - Vcc (±5V) and GND.
  - 3 x Enable pins.

Adobe Acrobat 7.0 Document

74LS138 3-8 Line Decoder





74LS138 Truth Table

$\overline{E}_1$	$\overline{E}_2$	$\overline{E}_3$	I_2	I_1	I_0		$\overline{O}_7$	$\overline{O}_6$	$\overline{O}_5$	$\overline{O}_4$	$\overline{O}_3$	$\overline{O}_2$	$\overline{O}_1$	$\overline{O}_0$
1	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	0	X	X	X		1	1	1	1	1	1	1	1
0	0	1	0	0	0		1	1	1	1	1	1	1	0
0	0	1	0	0	1		1	1	1	1	1	1	0	1
0	0	1	0	1	0		1	1	1	1	1	0	1	1
0	0	1	0	1	1		1	1	1	1	0	1	1	1
0	0	1	1	0	0		1	1	1	0	1	1	1	1
0	0	1	1	0	1		1	1	0	1	1	1	1	1
0	0	1	1	1	0		1	0	1	1	1	1	1	1
0	0	1	1	1	1		0	1	1	1	1	1	1	1

MAD Design using Decoder

1. Determine available information.
2. Determine the required number of address lines.
3. Set base address.
4. Determine lower address range.
5. Determine upper address range.
6. Design decoder.
7. Draw memory block diagram.

Example #1

Example #1

- Design a FAD using the 74LS139 2 to 4 decoder.
- The memory map is:
  - ROM1: 4kB, starting address $0000.
  - ROM2: 4kB, starting address $1000.
  - RAM1: 4kB, starting address $2000.

Step 1: Determine Available Information

- 12 kB need to be interfaced:
  - 6 chips used.
  - ROM1: 2kB even, 2kB odd.
  - ROM2: 2kB even, 2kB odd.
  - ROM2: 2kB even, 2kB odd.

Step 2: Determine Number of Required Address Lines (ROM1)

- Each chip contains 2,000 memory locations:
  - Needs 11 address lines.

$$2^x = 2{,}000$$

$$x \log_{10} 2 = \log_{10} 2{,}000$$

$$x = \frac{\log_{10} 2{,}000}{\log_{10} 2} = \frac{3.3010}{0.3010} = 10.97 \approx 11$$

Step 2: Determine Number of Required Address Lines (ROM2)

- Each chip contains 2,000 memory locations:
  - Needs 11 address lines.

$$2^x = 2{,}000$$

$$x \log_{10} 2 = \log_{10} 2{,}000$$

$$x = \frac{\log_{10} 2{,}000}{\log_{10} 2} = \frac{3.3010}{0.3010} = 10.97 \approx 11$$

Step 2: Determine Number of Required Address Lines (RAM1)

- Each chip contains 2,000 memory locations:
  - Needs 11 address lines.

$$2^x = 2{,}000$$

$$x \log_{10} 2 = \log_{10} 2{,}000$$

$$x = \frac{\log_{10} 2{,}000}{\log_{10} 2} = \frac{3.3010}{0.3010} = 10.97 \approx 11$$

Step 3,4,5: ROM1

A23	A22	A21	A20	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	X	X	X	X	X	X	X	X	X	X	X	
0				0				0															

A23	A22	A21	A20	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				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1
0				0				0				F				F				F			

= to decoder

Step 3,4,5: ROM2

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	
0				0				1															

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				1				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				1				F				F				F			

= to decoder

Step 3,4,5: RAM1

A23	A22	A21	A20	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	1	0	X	X	X	X	X	X	X	X	X	X	X	
0				0				2															

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				2				0				0				0			

A23	A22	A21	A20	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	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0				0				2				F				F				F			

= to decoder

Step 6: Design Decoder

- For 74LS139, two address lines need to be selected as the input for the decoder.
- Select two unique lines that identify ROM1, ROM2 and RAM1.
- Since this is a FAD, all of the remaining address lines must be used to generate the enable (E) signal.

Step 6: Design Decoder

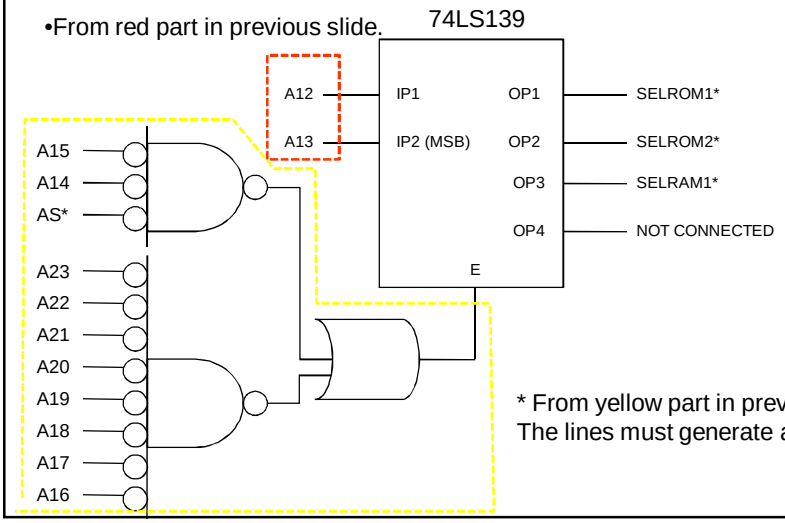
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	
0				0				0														ROM1	

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	
0				0				1														ROM2	

A23	A22	A21	A20	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	1	0	X	X	X	X	X	X	X	X	X	X	X	
0				0				2														RAM1	

Step 6: Design Decoder

•From red part in previous slide.



* From yellow part in previous slide.
The lines must generate a 0 at E.

Example #2

Example #2

- Design a PAD to implement the following memory map:
  - ROM1: $5000 - $5FFF.
  - RAM1: $A000 - $AFFF
- Use the 74LS138 3-8 decoder.

ROM1: $5000 to $5FFF

A23	A22	A21	A20	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	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0				0				5				0				0				0			

A23	A22	A21	A20	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	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				5				F				F				F			

A23	A22	A21	A20	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	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM1: $5000 to $5FFF

A23	A22	A21	A20	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	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				A				0				0				0			

A23	A22	A21	A20	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	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0				0				A				F				F				F			

A23	A22	A21	A20	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	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

Step 6: Design Decoder

- For 74LS138, three address lines need to be selected as the input for the decoder.
- Select three unique lines that identify ROM1 and RAM1.
- Since this is a PAD, some of the remaining address lines must be used to generate the enable (E) signal.

Step 6: Design Decoder

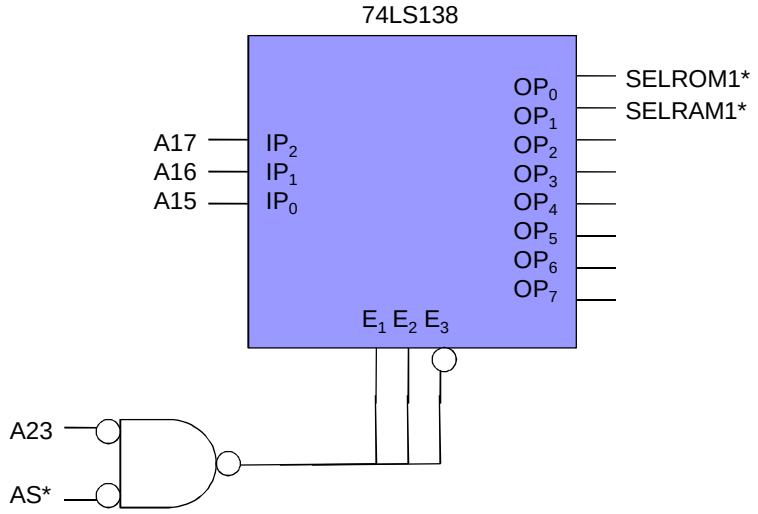
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0				0				A				F				F				F			

RAM1

A23	A22	A21	A20	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	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
0				0				5				F				F				F			

ROM1

Step 6: Design Decoder



Example #3

Example #3

- A 256kB RAM memory is composed of sixteen 16kB RAM chips. The address ranges for the RAM chips are as follows:
  - 00000 to 07FFF
  - 08000 to 0FFFF
  - 10000 to 17FFF
  - 18000 to 1FFFF
  - 20000 to 27FFF
  - 28000 to 2FFFF
  - 30000 to 37FFF
  - 38000 to 3FFFF
- Use the 74LS138 to design the FAD.

RAM1: $00000 to $07FFF

A23	A22	A21	A20	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				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM2: $08000 to $0FFFF

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM3: $10000 to $17FFF

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM4: $18000 to $1FFFF

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM5: $20000 to $27FFF

A23	A22	A21	A20	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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM6: $28000 to $2FFFF

A23	A22	A21	A20	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	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM7: $30000 to $37FFF

A23	A22	A21	A20	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	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

RAM8: $38000 to $3FFFF

A23	A22	A21	A20	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	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				0				0				0			

A23	A22	A21	A20	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	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0				0				7				F				F				F			

A23	A22	A21	A20	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	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

= to decoder

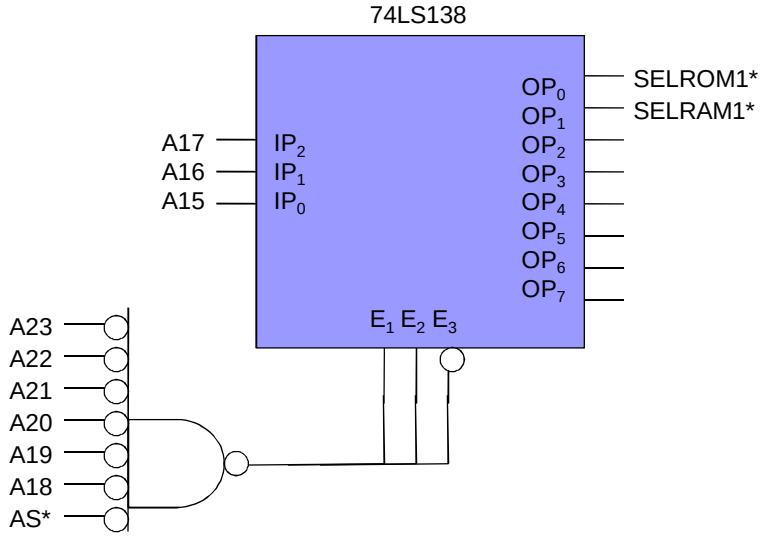
RAM1 – RAM8

A23	A22	A21	A20	A19	A18	A17	A16	A15
0	0	0	0	0	0	0	0	0
						0	0	1
						0	1	0
						0	1	1
						1	0	0
						1	0	1
						1	1	0
						1	1	1

Goes to activate Decoder (E1,E2,E3)

Goes to decoder input (IP0,IP1,IP2)

Step 6: Design Decoder



Example #4

Example #4

- Implement a PAD for the following memory map:
  - RAM: $400400 - $4004FF.
  - I/O1: $400500 - $40053F.
  - I/O2: $400540 - $40057F.
- Use the 74LS139 decoder.

Step 1: Determine Available Information

- 3 types of memory locations need to be decoded.
- Memory size not given.
- Address range given:
  - RAM: $400400 - $4004FF.
  - I/O1: $400500 - $40053F.
  - I/O2: $400540 - $40057F.

Step 2a: Determine Number of Required Address Lines (RAM)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4				0				0				4				0				0			

A23	A22	A21	A20	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	1	0	0	1	1	1	1	1	1	1	1
4				0				0				4				F				F			

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

Step 2b: Determine Number of Required Address Lines (I/O1)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
4				0				0				5				0				0			

A23	A22	A21	A20	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	1	0	1	0	0	1	1	1	1	1	1
4				0				0				5				3				F			

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

Step 2b: Determine Number of Required Address Lines (I/O2)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
4				0				0				5				4				0			

A23	A22	A21	A20	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	1	0	1	0	1	1	1	1	1	1	1
4				0				0				5				7				F			

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Step 7: Find Unique Pattern

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

A23	A22	A21	A20	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	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Try to cancel out, there are not unique patterns!

Step 7: Find Unique Pattern

- To design a decoder for these types of memory maps, the decoder has to be designed in two stages.
- The first stage is to design the decoder for RAM, I/O1 and I/O2.
- The second stage is to design the decoder again for I/O1 and I/O2.

Step 7: Find Unique Pattern

A23	A22	A21	A20	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	1	0	0	X	X	X	X	X	X	X	X
4				0				0				4											

A23	A22	A21	A20	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	1	0	1	0	0	X	X	X	X	X	X
4				0				0				5											

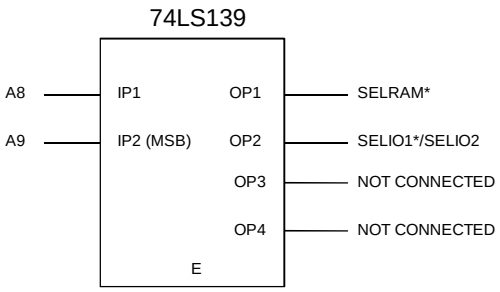
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	X	X	X	X	X	X
4				0				0				5											

Step 8: Design Decoder Circuit (Stage 1)

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	X	X	X	X	X	X	X	X

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	X	X	X	X	X	X

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	1	X	X	X	X	X	X



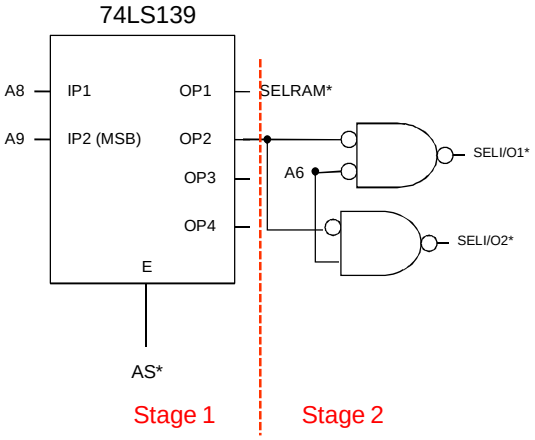
Step 8: Design Decoder Circuit (Stage 2)

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	X					X	X	X

DONE

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	X	X	X	X	X	X

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	1	X	X	X	X	X	X



Example #5

Example #5

- Implement the following memory map using the 74LS138 decoder:
  - ROM1: $042000 - $0427FF.
  - RAM1: $800000 - $8007FF.
  - I/O1: $043080 - $0430BF.
  - I/O2: $043040 - $04307F.
- Use the partial decoding method.

Step 2a: Determine Number of Required Address Lines (ROM1)

A23	A22	A21	A20	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	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0				4				2				0				0				0			

A23	A22	A21	A20	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	1	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
0				4				2				7				F				F			

A23	A22	A21	A20	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	1	0	0	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X

Step 2b: Determine Number of Required Address Lines (RAM1)

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8				0				0				0				0				0			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
8				0				0				7				F				F			

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X

Step 2c: Determine Number of Required Address Lines (I/O1)

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
0				4				3				0				8				0			

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	1	0	1	1	1	1	1	1
0				4				3				0				B				F			

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	1	0	X	X	X	X	X	X

Step 2d: Determine Number of Required Address Lines (I/O2)

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
0				4				3				0				4				0			

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1
0				4				3				0				7				F			

A23	A22	A21	A20	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	1	0	0	0	0	1	1	0	0	0	0	0	1	X	X	X	X	X	X

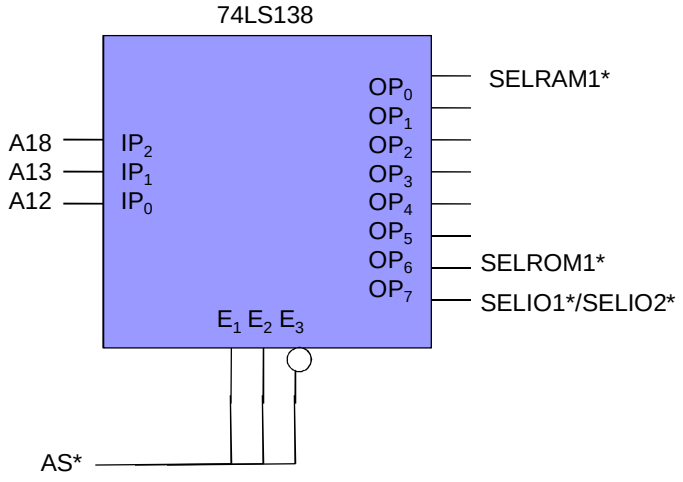
Find Unique Pattern

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6
0	0	0	0	0	1	0	0	0	0	1	0	0	X	X	X	X	X
1	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X
0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1

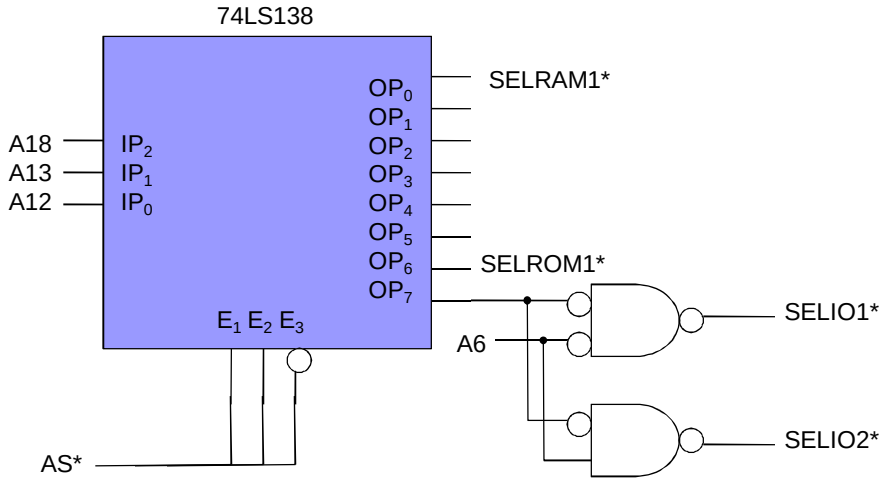
Find Unique Pattern

A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6
0					1					1	0		X	X	X	X	X
1					0					0	0		X	X	X	X	X
0					1					1	1					1	0
0					1					1	1					0	1

Design Decoder: Stage 1



Design Decoder: Stage 2



Conclusion

Memory Chips

- Has:
  - Address pins.
  - Data pins.
  - CS*
  - OE*/E*
  - WE* (RAM only).
- Both E* and CS* must be on to activate a memory chip.
- E* is activated by UDS*/LDS*.
- CS* is activated by memory address decoder.

Memory Interfacing

- Method to connect memory chips to M68k.
- Involves design of MAD.
- Two methods:
  - Full address decoding: uses all address lines.
  - Partial address decoding: only uses several address lines.
- Can be designed using logic gates and decoder ICs.

The End

Please read:

Antonakos, 263-275

Extra Reading

Bus Buffering

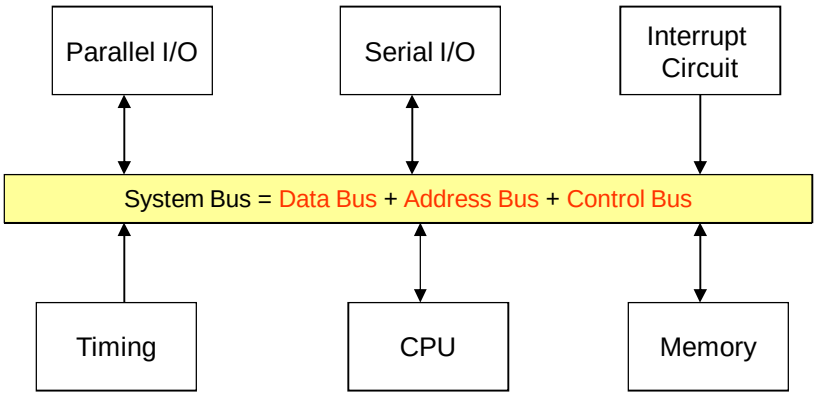
Bus Buffering

- The system bus consists of:
  - Address bus: A1 – A23
  - Data bus: D0 – D15
  - Control bus: $\overline{BERR}$, $\overline{VPA}$, CLK, R/$\overline{W}$, etc..

Bus Buffering

- System bus is connected to all components in μP system:
  - Memory: RAM, ROM.
  - I/O devices: keyboard, mouse, display card.
  - Storage devices: HDD, CD-ROM drive.

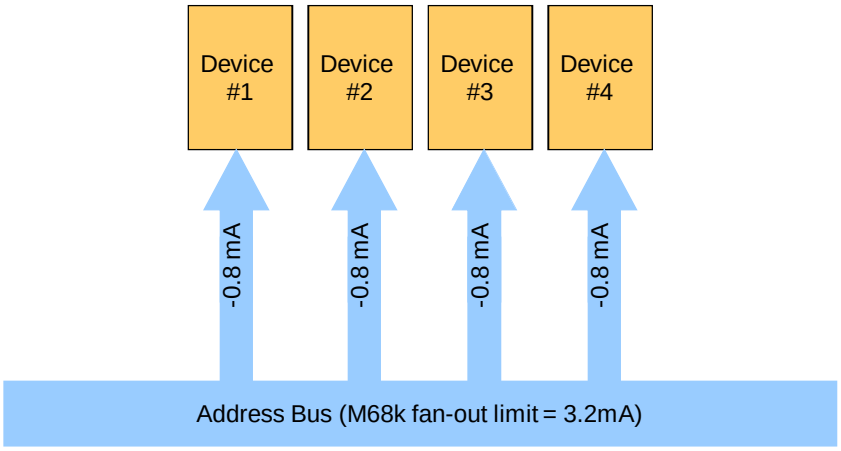
Block Diagram



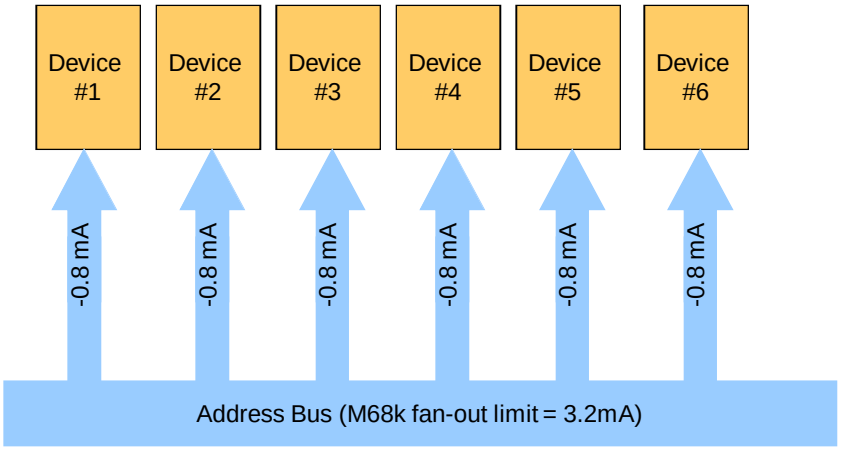
Fan-Out

- All connected devices drain current from M68k.
- This is called fan-out.
- If too many devices connected, M68k fan-out overloaded:
  - Causes unreliable input/output.
  - Fan-out limit is specified by manufacturer.

Fan-Out: Normal Situation



Fan-Out: Overloaded Situation



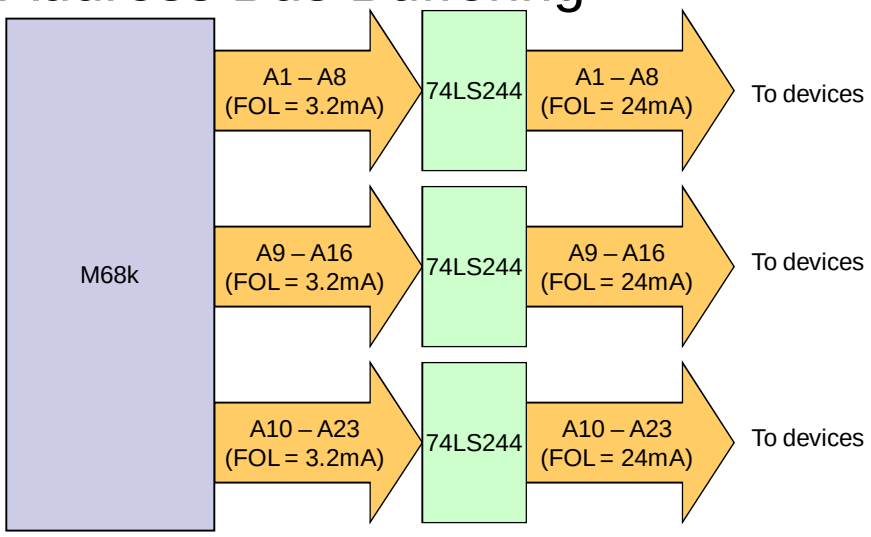
Bus Buffering

- Buffering helps prevent bus overloading by increasing the bus fan-out current.
  - 2 buffer choices 74LS244 or 74LS245.
  - Depends on direction:
    - Unidirectional
    - Bidirectional

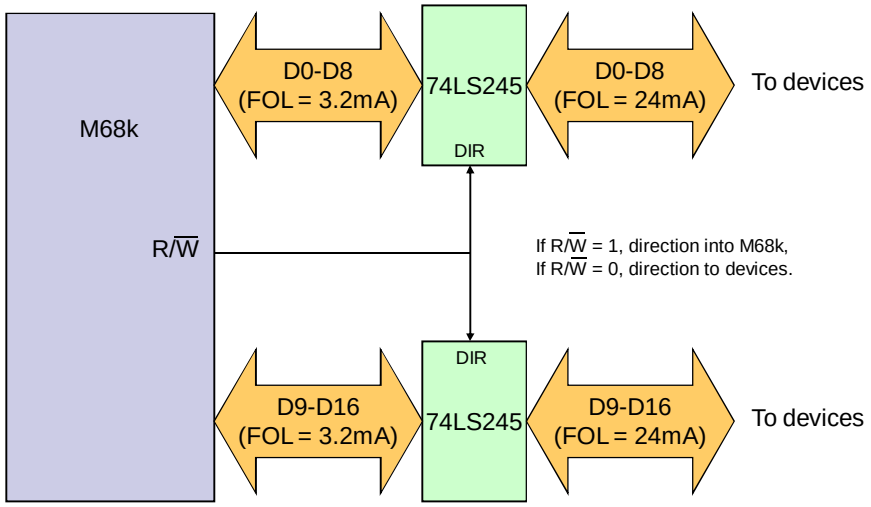
Bus Buffering

- Increases M68k fan-out limit by attaching high-current buffer to system bus.
  - Can connect more devices.
- Common buffers:
  - 74LS244 unidirectional buffer: can transfer data in one direction only.
  - 74LS245 bidirectional buffer: can transfer data in both directions.
  - Each IC can buffer 8 lines only.

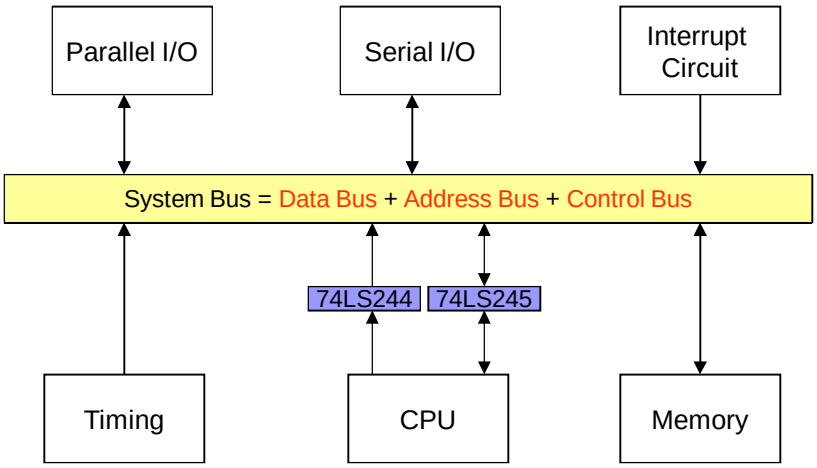
Address Bus Buffering



Data Bus Buffering

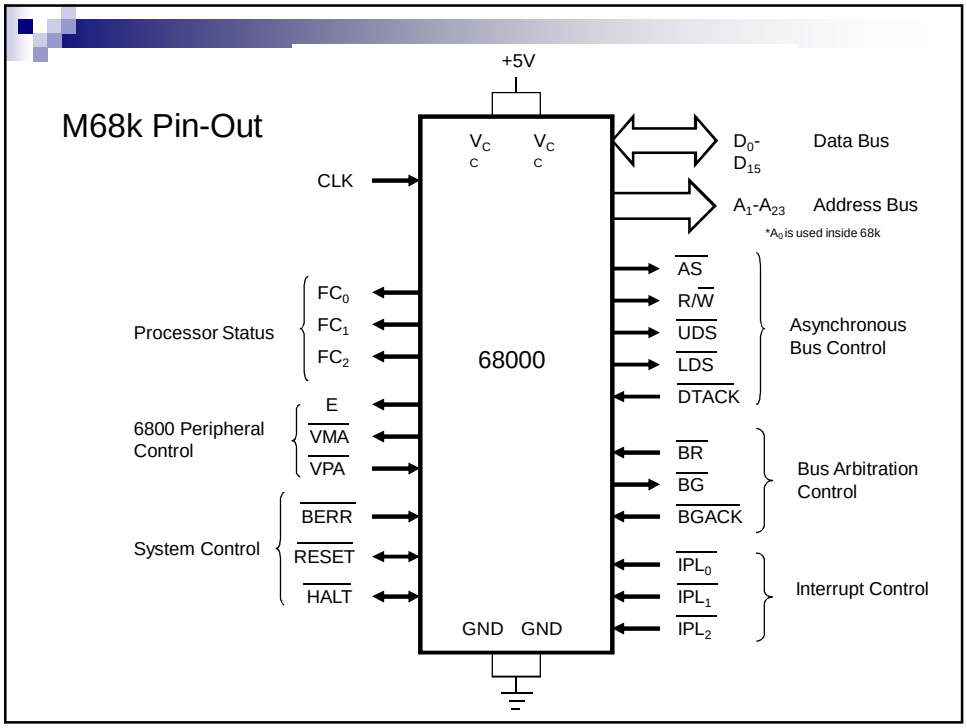


Buffering in M68k System



Bus Buffering

- All pins in M68k need to be buffered:
  - Address bus: buffered using 74LS244.
  - Data bus: buffered using 74LS245.
  - Control bus: buffered using 74LS244/74LS245:
    - If unidirectional, use 74LS244.
    - If bidirectional, use 74LS245.


M68k Pin-Out
+5V
VCC
VCC
CLK
68000
GND GND
D0-D15
Data Bus
A1-A23
Address Bus
*A0 is used inside 68k
AS
R/W
UDS
LDS
DTACK
Asynchronous Bus Control
Processor Status
FC0
FC1
FC2
6800 Peripheral Control
E
VMA
VPA
System Control
BERR
RESET
HALT
BR
BG
BGACK
Bus Arbitration Control
IPL0
IPL1
IPL2
Interrupt Control