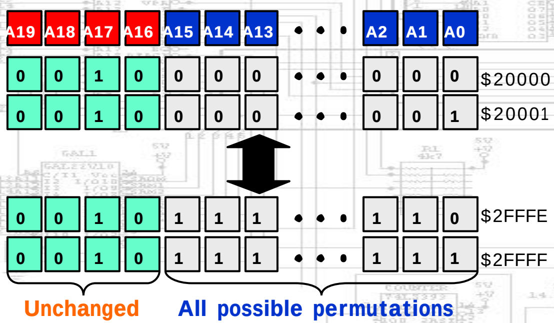
2BA4

Example: RAM$_2$



2BA4

Number of address lines decoded

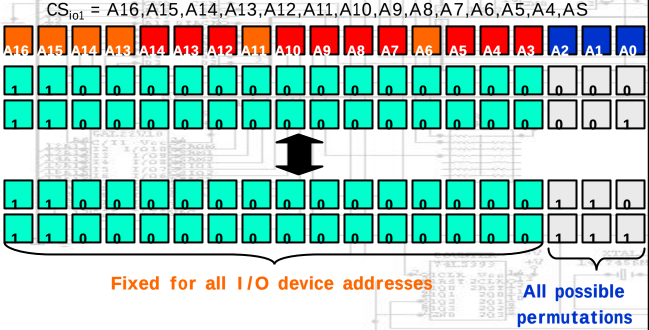
No of bits decoded =
Total address bus width -
Number of address lines into device.

2BA4

I/O Device$_1$ (\$0000... \$0007)

16 address lines ~~A15–A0~~

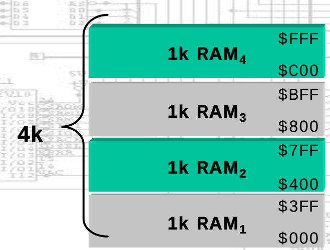
CS_{io1} = A16,A15,A14,A13,A12,A11,A10,A9,A8,A7,A6,A5,A4,AS



2BA4

Good Memory Map Guidelines

- 1k RAM -> 1k = 2^{10}
- Mapped into a system with 12 address lines (4k)



Implement CS logic for $RAM_1 .. RAM_4$

	A11	A10	A9	A8	A7	• • •	A2	A1	A0	
RAM_1	0	0	0	0	0	• • •	0	0	0	$000
	0	0	1	1	1	• • •	1	1	1	$3FF
RAM_2	0	1	0	0	0	• • •	0	0	0	$400
	0	1	1	1	1	• • •	1	1	1	$7FF
RAM_3	1	0	0	0	0	• • •	0	0	0	$800
	1	0	1	1	1	• • •	1	1	1	$BFF
RAM_4	1	1	0	0	0	• • •	0	0	0	$C00
	1	1	1	1	1	• • •	1	1	1	$FF

Used for decoding (A11, A10)

Address bits fed into RAM chips (A9 … A0)

Device Base Address

- Start device address ranges at addresses that are multiples of corresponding devices sizes.
  - **Base(d) = n * size(d)**
    - Where n is 0, 1, 2, 3…
  - Guarantees fixed pattern of high address bits .
  - Simple decoding

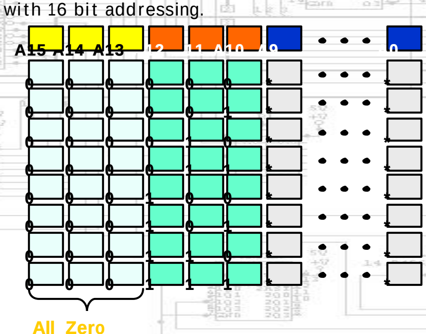
Example:

- A 1MB device should start at an address of the form $XXX00000.
- 2k = $800
  - Valid base addresses for 2k devices:
    - XX000 and XX800
- 8k = $2000
  - Valid base addresses for 8k devices:
    - X000, X2000, X4000, X6000, X8000, A000, C000, E000

Exploiting Common Bit Pattern

Many devices with common high order address pattern.
8 x 1k Ram devices with 16 bit addressing.

RAM_1=$0000-$03FF
RAM_2=$0400-$07FF
RAM_3=$0800-$0BFF
RAM_4=$0C00-$0FFF
RAM_5=$1000-$13FF
RAM_6=$1400-$17FF
RAM_7=$1800-$1BFF
RAM_8=$1C00-$1FFF



2BA4

3-to-8 Address Decoder