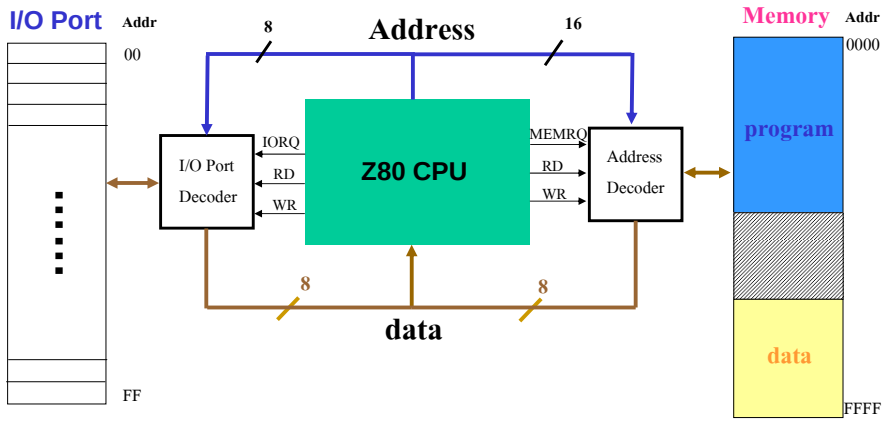
Chapter 5

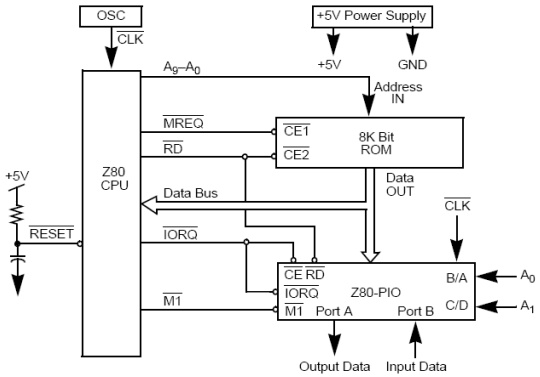
Z80 Computer System Design

Simplified Z80 System Architecture



Minimum Z80 Computer System

- Consisting of Z80CPU, Oscillator, ROM, I/O
- Additional RAM is preferred
- Total starting cost for DIY ~500 THB
- Complete and ready to use board ~1,000 THB



Minimum Z80 Computer System

- Z80CPU + RAM + ROM (Socket) + I/O Controller + Oscillator
- System extension via system bus connector

Main memory

- Memory
  - Stores programs
  - Provides data to the MPU
  - Accepts result from the MPU for storage
- Main memory Types
  - ROM : read only memory. Contains program (Firmware). does not lose its contents when power is removed (Non-volatile)
  - RAM: random access memory (read/write memory) used as variable data, loses contents when power is removed volatile. When power up will contain random data values

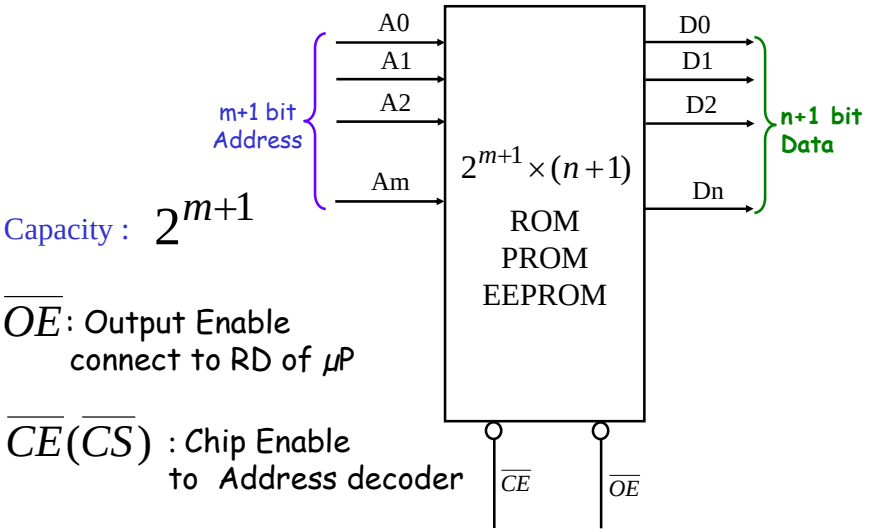
Read-Only Memory (ROM)

- µP can read instructions from ROM quickly
- Cannot write new data to ROM
- ROM stores data, even after power cycled
- When power is turned on, the microprocessor will start fetching instructions from ROM (bootstrap)

Available ROMs

- Masked ROM or just ROM
- PROM or programmable ROM (one-time programmable)
- EPROM (Erasable/Programmable ROM)
  Ultra Violet (UV) light is used in erasing process
- Flash
  - re-writable about 10,000 times
  - usually must write a whole block not just 1 or 2 bytes,
  - slow writing fast reading
- EEPROM (electrically erasable ROM)
  - fast writing slow reading
  - can program millions of times
  - useless for storing a program
  - good for save configuration information.

ROM

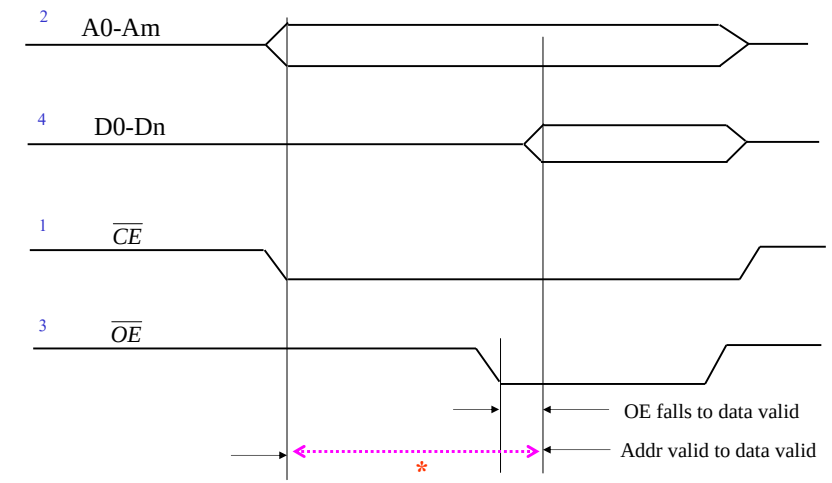


Capacity : 2^{m+1}

$\overline{OE}$: Output Enable
connect to RD of µP

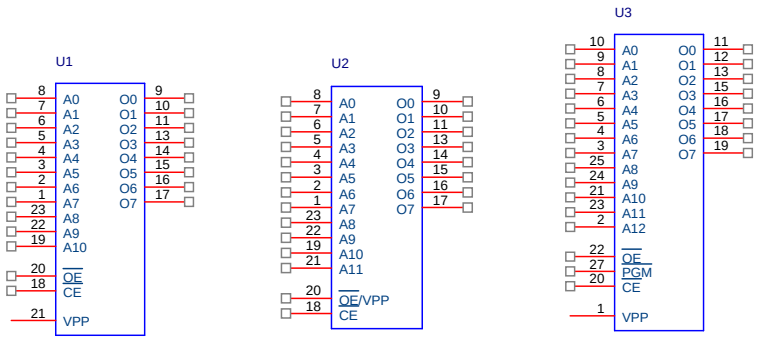
$\overline{CE}(\overline{CS})$: Chip Enable
to Address decoder

ROM Read Timing



* Defined by manufacturer

27XX EPROM



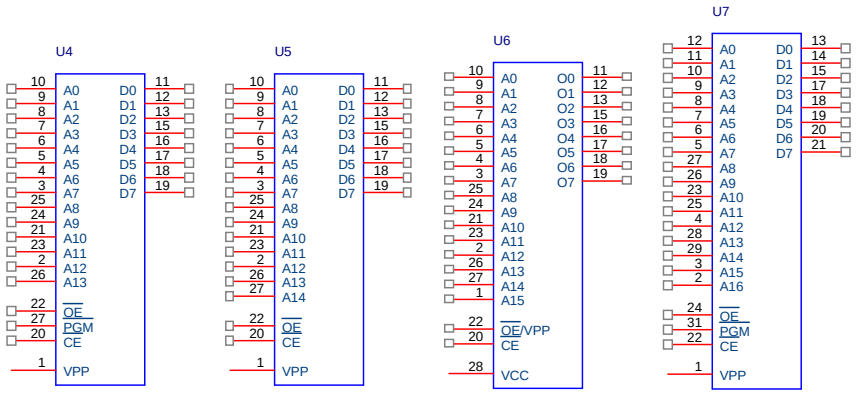
2716
16 kbit
2 kbyte

2732
32 kbit
4 kbyte

2764
64 kbit
8 kbyte

PGM and VPP are used to programming

27XXX EPROM



27128
128 kbit
16 kbyte

27256
256 kbit
32 kbyte

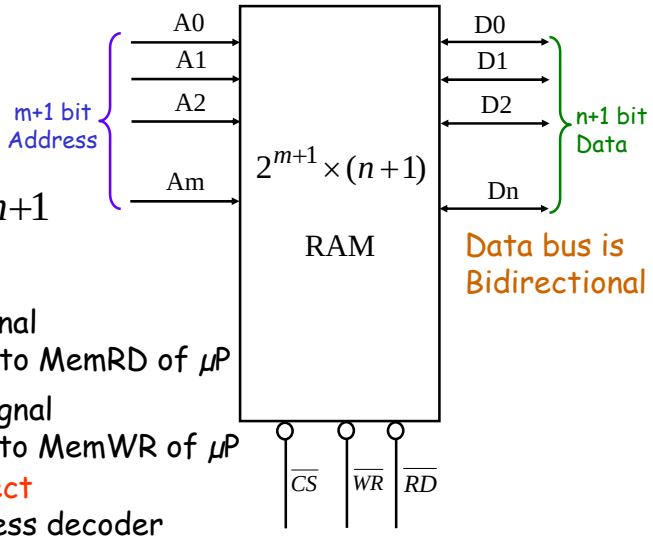
27512
512 kbit
64 kbyte

27010
1024 kbit
128 kbyte

RAM (Random Access Memory)

- µP can read the data from RAM quickly
- µP can write new data to RAM quickly
- RAM is unable to store data if power is turned off
- Two type is available :
  - Static RAM (SRAM): FF base, fast, expensive, low cap/vol, applied for cache , no refresh
  - Dynamic RAM (DRAM): capacitor base, slow , low cost high capacity/volume , applied for main memory(pc) need refresh.

RAM(Static)



Capacity : 2^{m+1}

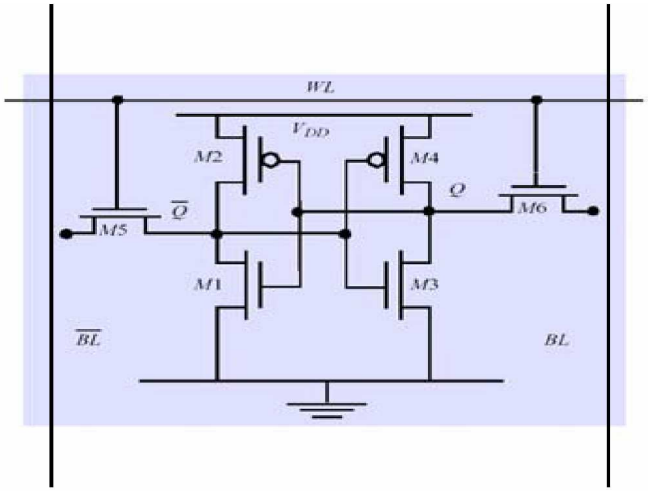
Data bus is Bidirectional

$\overline{RD}$: Read signal
connect to MemRD of μP

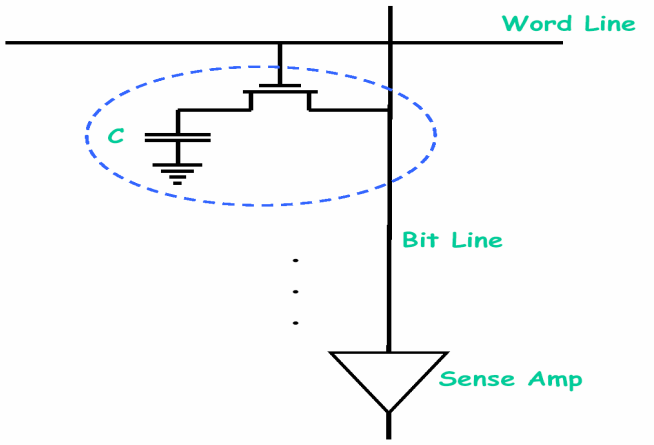
$\overline{WR}$: Write signal
connect to MemWR of μP

$\overline{CS}$: Chip Select
to Address decoder

Static RAM

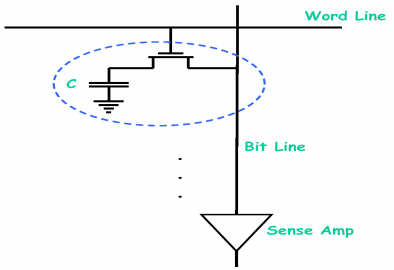


Dynamic RAM



Dynamic RAM

- **Write :** Charge bit line HIGH or LOW and set word line HIGH
- **Read :** Bit line is precharged to a voltage halfway between HIGH and LOW and then the word line is set HIGH.
- Sense Amp Detects change
- Reads are destructive (Must follow with a write)
- Address Buffer

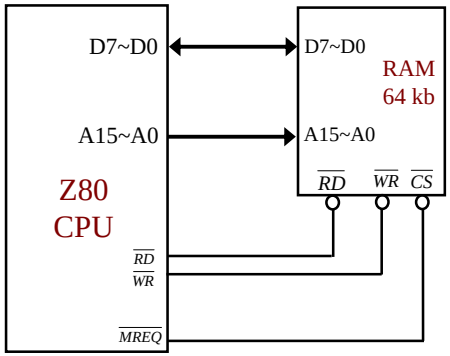


Z80 Memory connection

- CPU 16 bit address bus ➔ 64 k memory(max)
- CPU 8 bit data bus ➔ 8 bit data width
- Generally should be connected
  - Data to data
  - Address to address
  - WR to wr
  - RD to rd
  - MREQ to cs

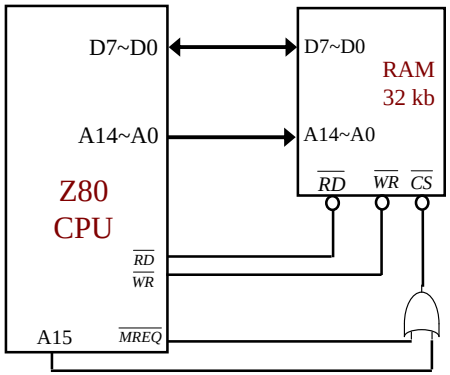
Memory connection (cont.)

❑ If only one RAM chip with Full size (64 kb capacity)



Memory connection (cont.)

❑ If RAM capacity is 32 kb
❑ A15 is combined with MREQ
❑ RAM address is from 0000h to 7FFFh

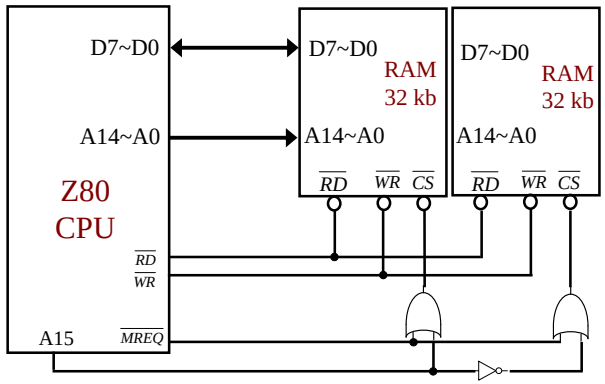


Memory connection (cont.)

- Given two 32K chips, how to obtain full 64K RAM address?
- Problem: Bus Conflict. The two memory chips will provide data at the same time when microprocessor performs a memory read.
- Solution: Use address line A15 as an “arbiter”. If A15 outputs a logic “1” the upper memory is enabled (and the lower memory is disabled) and vice-versa.

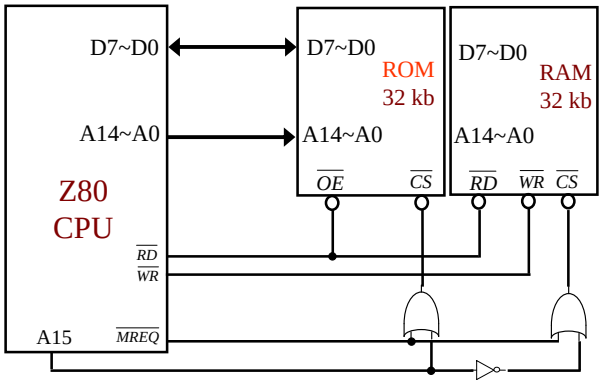
Memory connection (cont.)

- There are two 32K RAM
- A15 applied to select one RAM chip
- Two RAM area is from 0000h to 7FFFh
  and 8000h to FFFFh



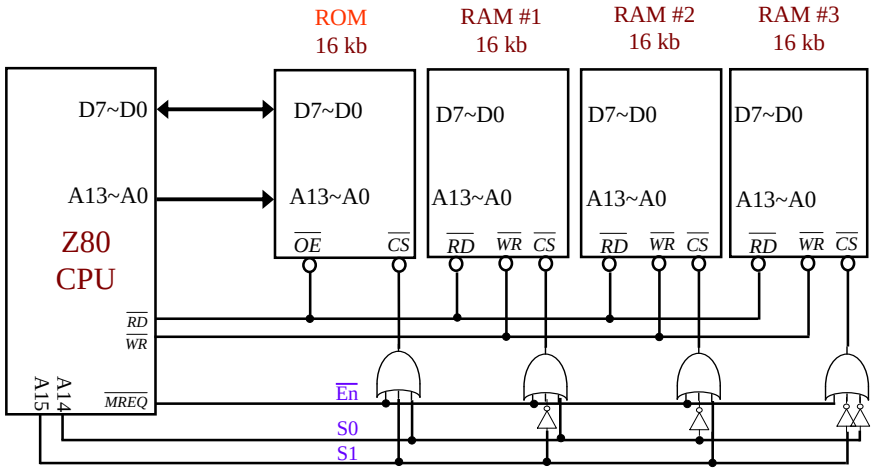
Memory connection (cont.)

- 32K ROM and 32K RAM
- ROM doesn't have WR signal



Memory connection (cont.)

There are 4 memory chip
A14 and A15 are applied to chip selection



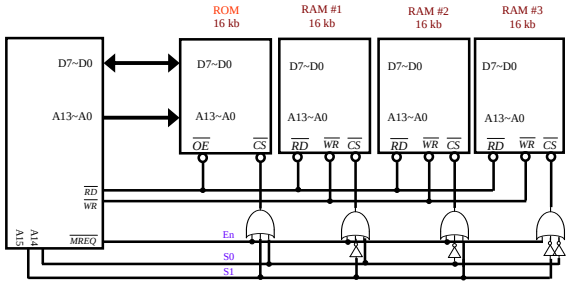
Address Bit Map

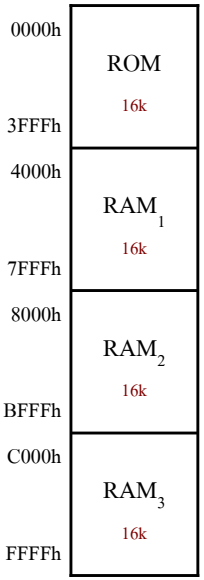
Selects chip — Selects location within chips

A15 to A0 (HEX)	AA 11 54	AA 11 32	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0000h	00	00	0000	0000	0000	ROM
3FFFh	00	11	1111	1111	1111	
4000h	01	00	0000	0000	0000	RAM_1
7FFFh	01	11	1111	1111	1111	
8000h	10	00	0000	0000	0000	RAM_2
BFFFh	10	11	1111	1111	1111	
C000h	11	00	0000	0000	0000	RAM_3
FFFFh	11	11	1111	1111	1111	

Memory Map

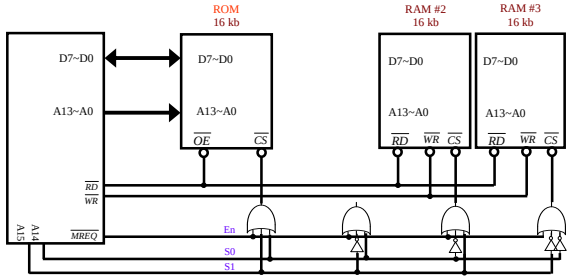
- Represents the memory type
- Address area of each memory chip
- Empty area

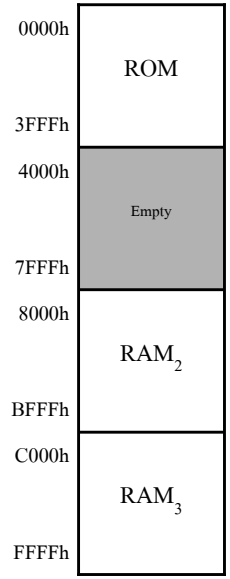




Memory Map

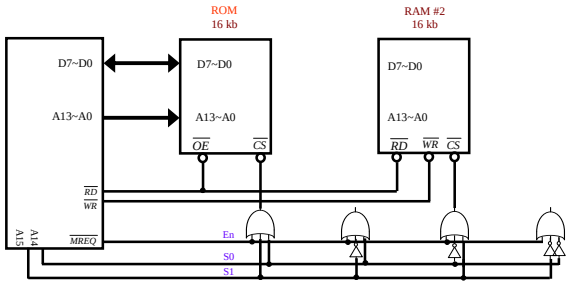
- Empty area is neither writable nor readable
- Read op. returns FFh value (usualy)
- Write op. can't store any value on it

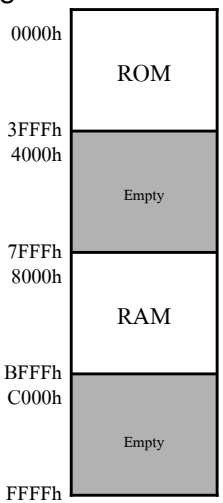




Memory Map

- Empty area is neither writable nor readable
- Read op. returns FFh value (usualy)
- Write op. can't store any value on it



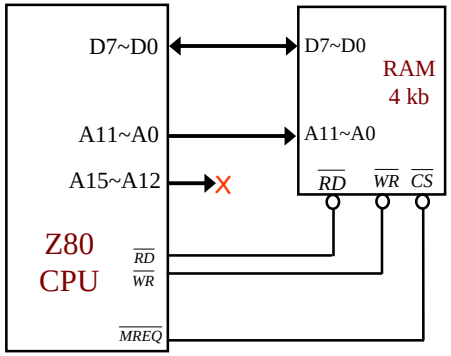


Full and Partial Decoding

- Full (exhaust) Decoding
  - All of the address lines are connected to any memory/device to perform selection
  - Absolute address : any memory location has one address
- Partial Decoding
  - When some of the address lines are connected the memory/device to perform selection
  - Using this type of decoding results into roll-over addresses (fold back or shading).
  - roll-over address : any memory location has more than one address

Partial Decoding

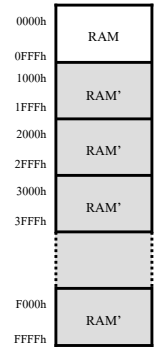
- A15~A12 are not connected
- What is the memory map?



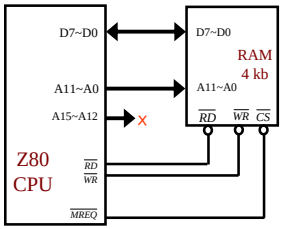
Partial Decoding

- Every memory location has more than one address
- For example first RAM location has addresses:
  - 0000h
  - 1000h
  - 2000h
  - 3000h
  - .....
  - .....
  - F000h

Roll-over Address

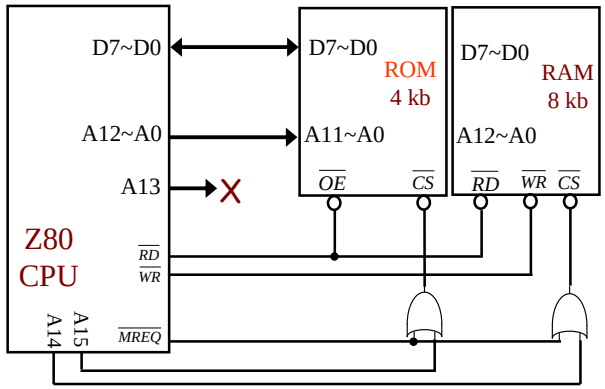


A15 to A0 (HEX)	AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
X000h	XXXX	0000	0000	0000	RAM
XFFFh	XXXX	1111	1111	1111	



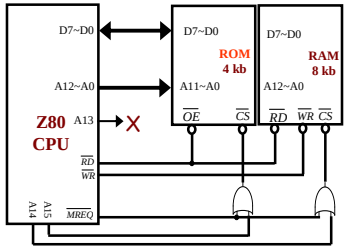
Partial Decoding

- A12 only connected to RAM
- A13 has no connection
- What is the memory map?



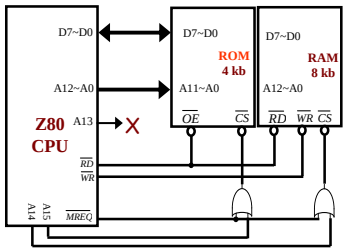
Partial Decoding

- 8 roll-over address for ROM
- 4 roll-over address for RAM

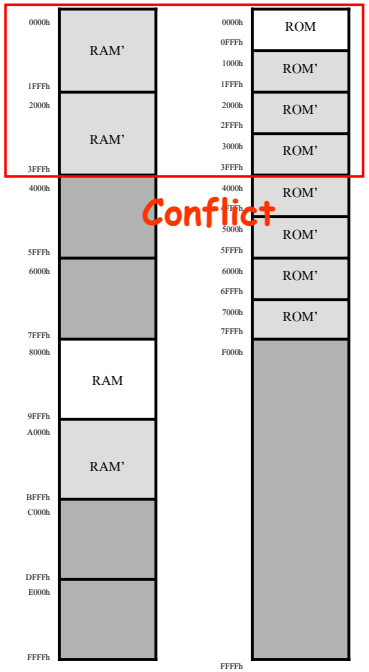


AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0xxx	0000	0000	0000	ROM
0xxx	1111	1111	1111	
X0x0	0000	0000	0000	RAM
X0x1	1111	1111	1111	

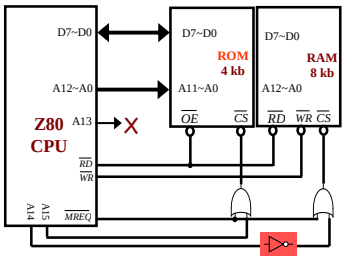
Partial Decoding



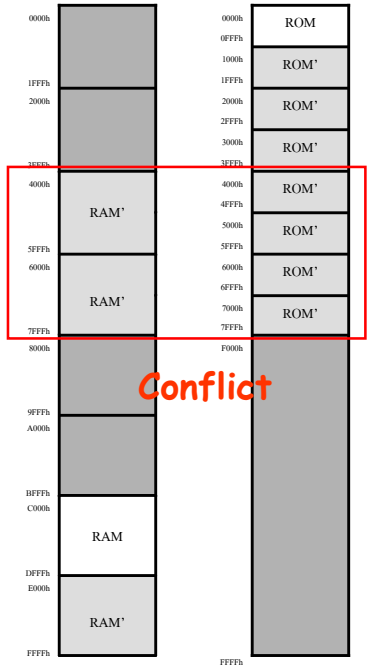
AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0xxx	0000	0000	0000	4k ROM
0xxx	1111	1111	1111	
X0x0	0000	0000	0000	8k RAM
X0x1	1111	1111	1111	



Partial Decoding

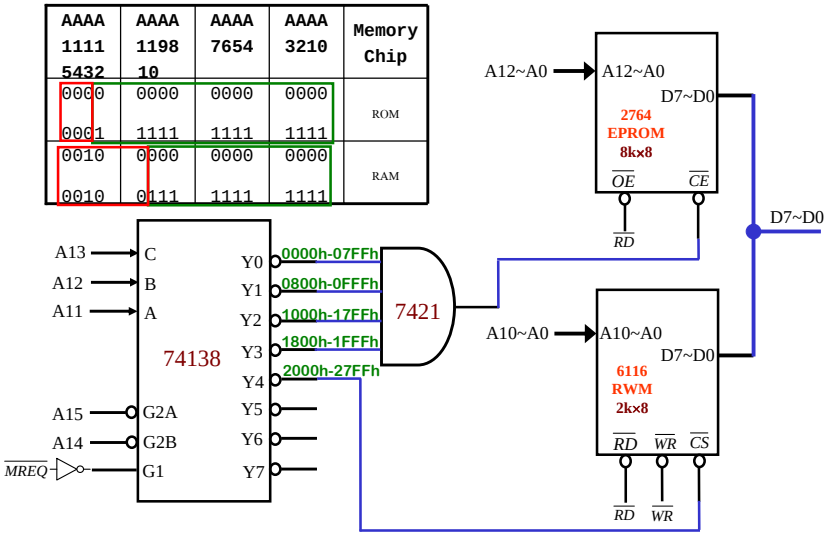


AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0xxx	0000	0000	0000	4k ROM
0xxx	1111	1111	1111	
X1x0	0000	0000	0000	8k RAM
X1x1	1111	1111	1111	



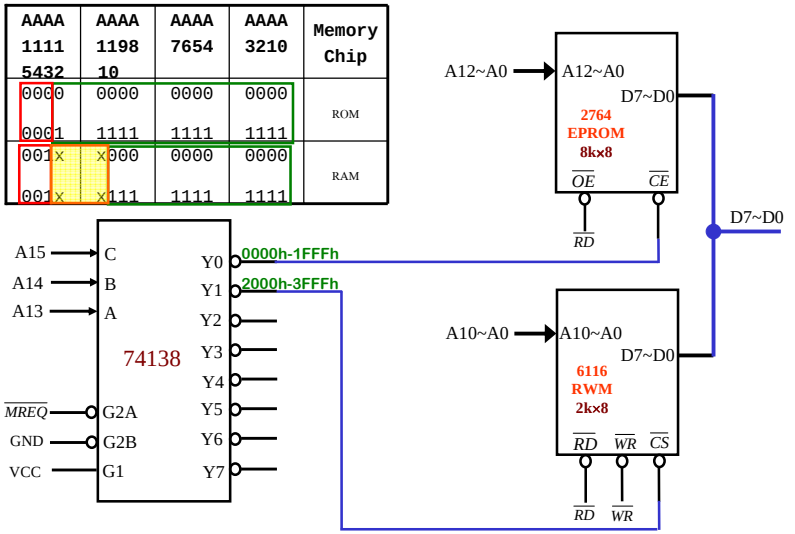
Full (exhaustive) decoding

AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0000	0000	0000	0000	ROM
0001	1111	1111	1111	
0010	0000	0000	0000	RAM
0010	0111	1111	1111	



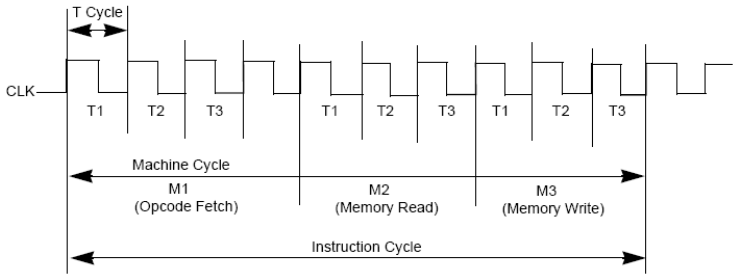
Partial decoding

AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0000	0000	0000	0000	ROM
0001	1111	1111	1111	
001x	x000	0000	0000	RAM
001x	x111	1111	1111	



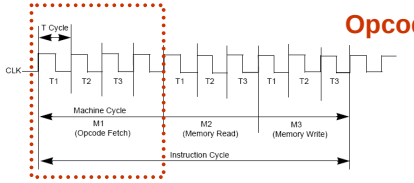
Z80 Timing (revisited)

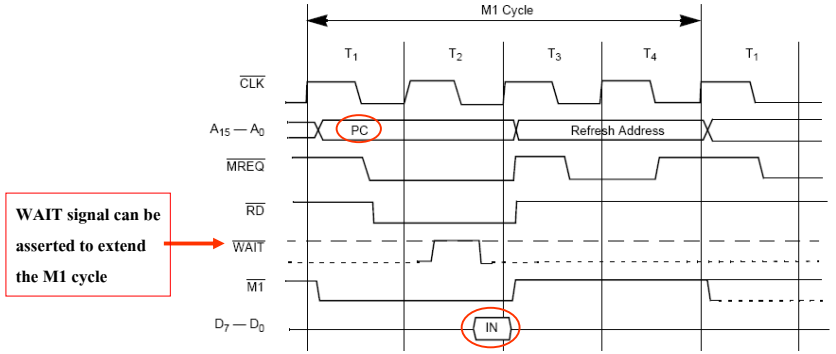
- The Z80 CPU executes instructions by stepping through a precise set of basic operations. These include:
  - **Memory Read or Write**
  - **I/O Device Read or Write**
  - **Interrupt Acknowledge**
- Three to six clock periods are required to complete each operations
- Number of clocks used in each operation can be extended to synchronize the CPU to the speed of external devices.



Z80 Timing (revisited)

Opcode Fetch

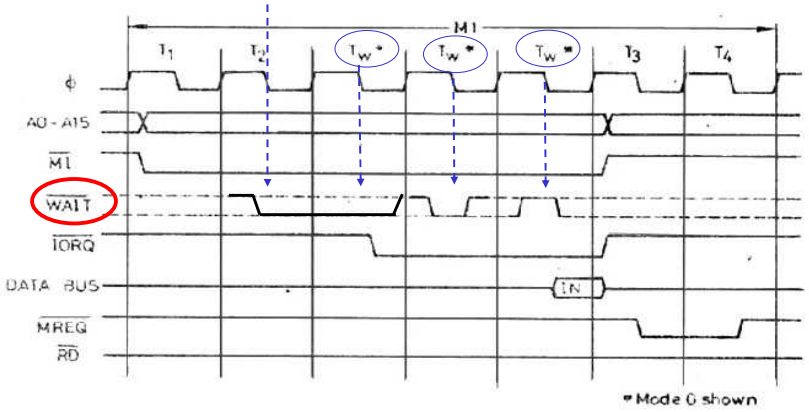




Z80 Timing (revisited)

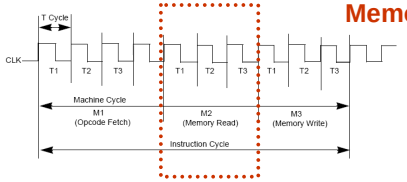
Wait State

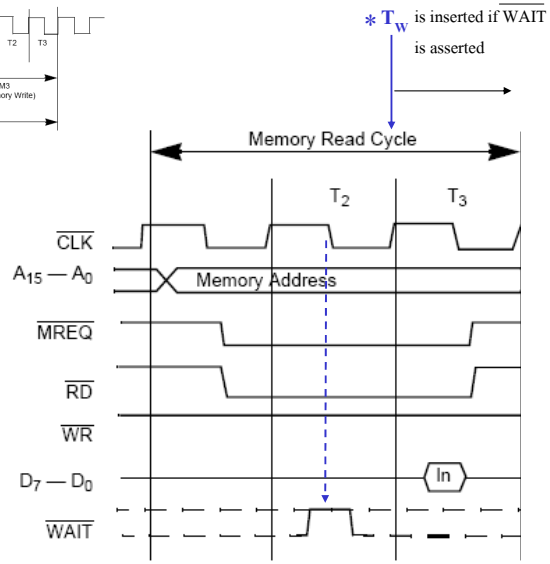
- WAIT signal is sampled at the falling edge of T2
- If it is asserted, wait state (Tw) is inserted until WAIT signal is removed



Z80 Timing (revisited)

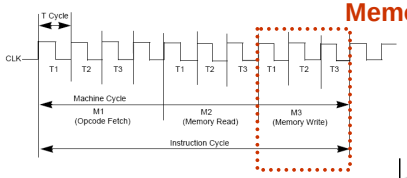
Memory Read

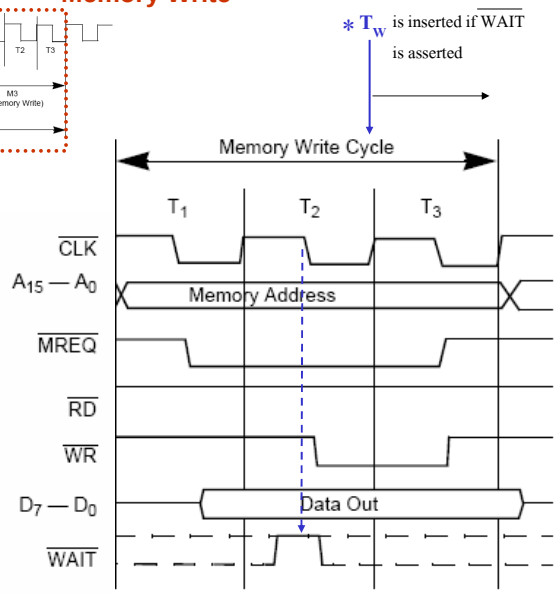




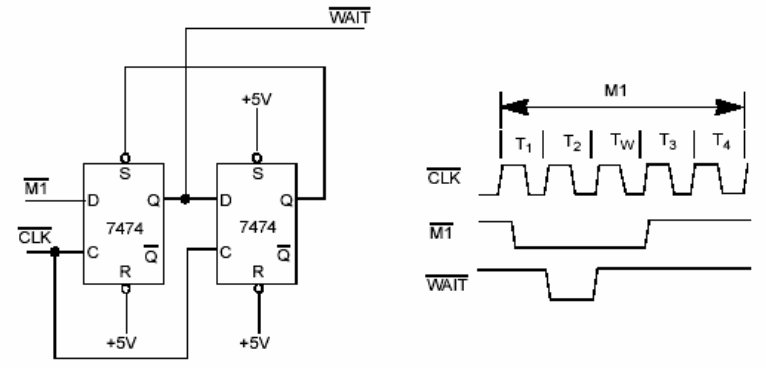
Z80 Timing (revisited)

Memory Write





Adding One Wait State to an M1 Cycle



Memory is a way of holding onto the things you love, the things you are, the things you never want to lose.

~From the television show *The Wonder Years*