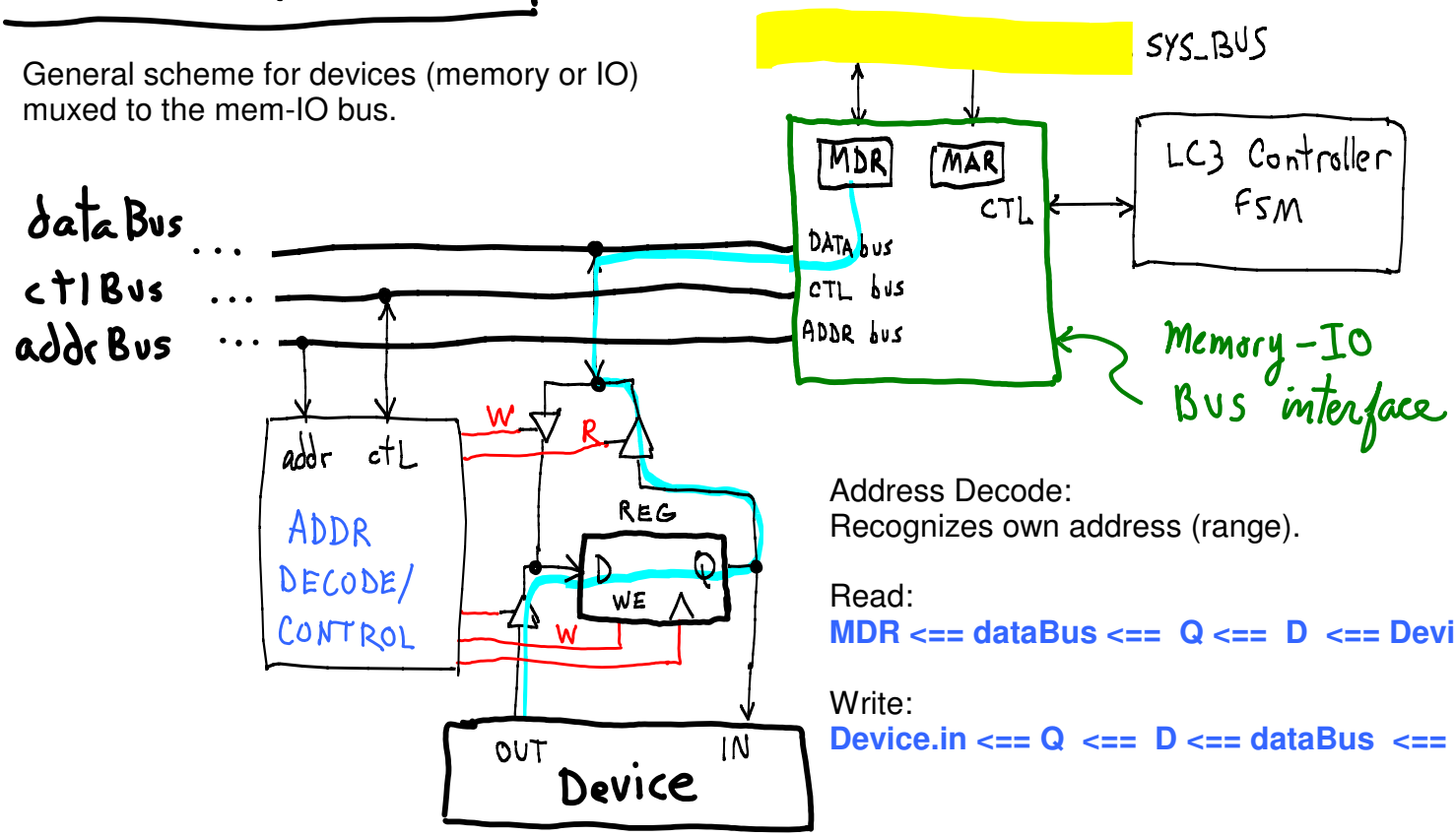
LC3 Memory, I/O Bus

General scheme for devices (memory or IO) muxed to the mem-IO bus.

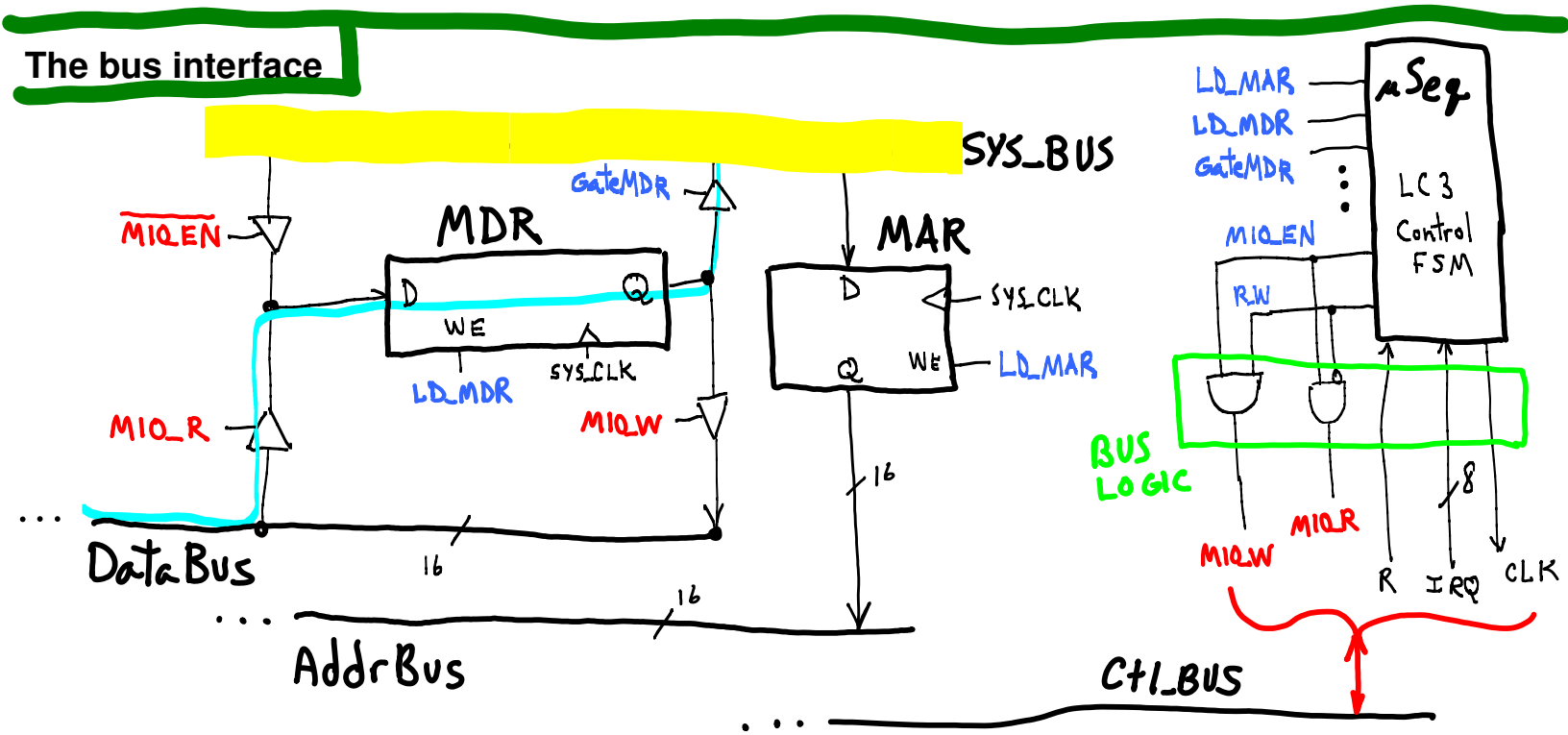


Address Decode:
Recognizes own address (range).

Read:
MDR <== dataBus <== Q <== D <== Device.out

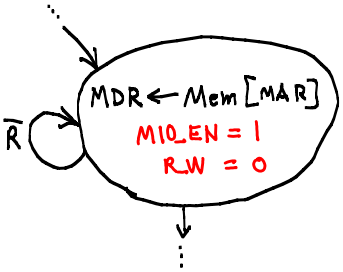
Write:
Device.in <== Q <== D <== dataBus <== MDR

The bus interface



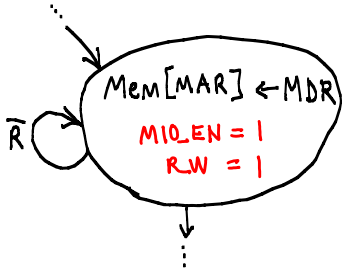
LC3 FSM
READ states:

33, 28, 24, 25, 29, 36, 40, 52*

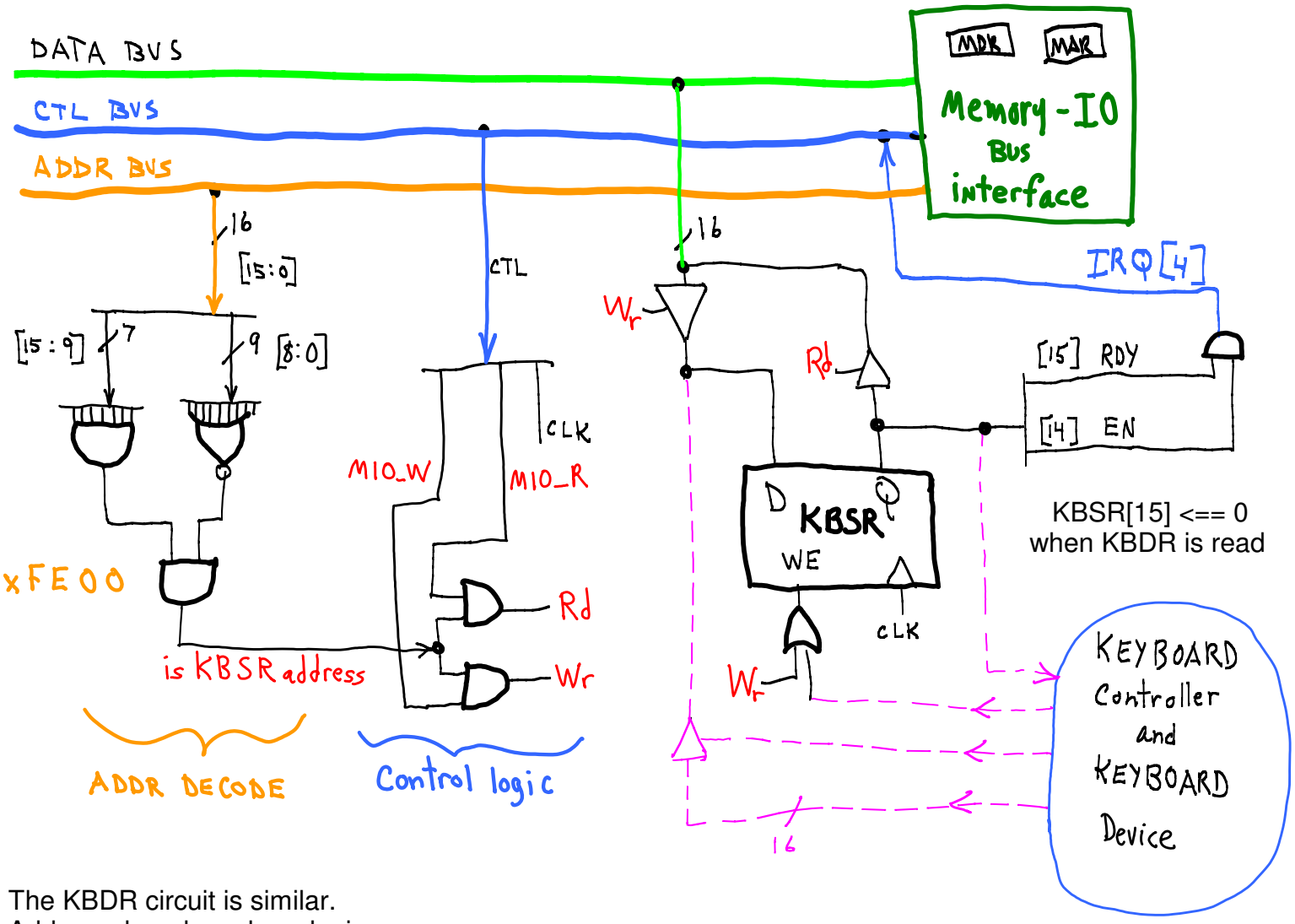


LC3 FSM
WRITE states:
16, 41*, 48*

* Interrupt mechanism not implemented in LC3simulate, lc3sim, or PennSim.



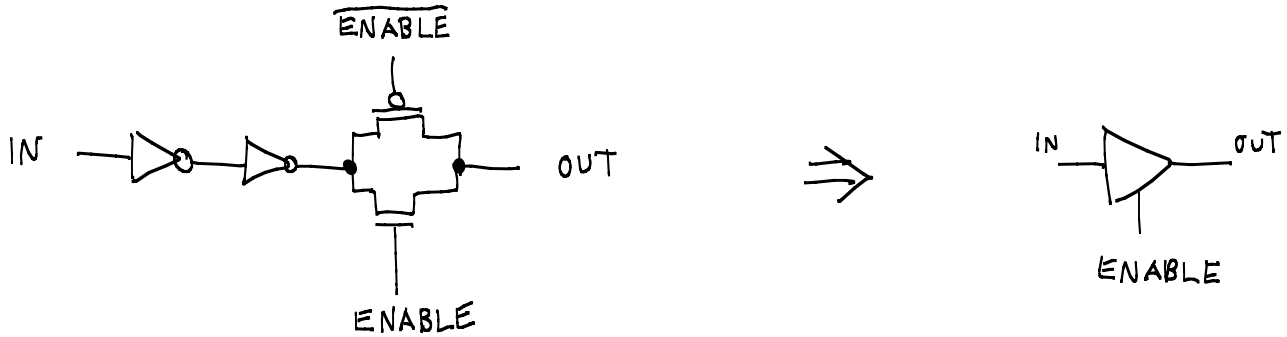
Keyboard, KBSR



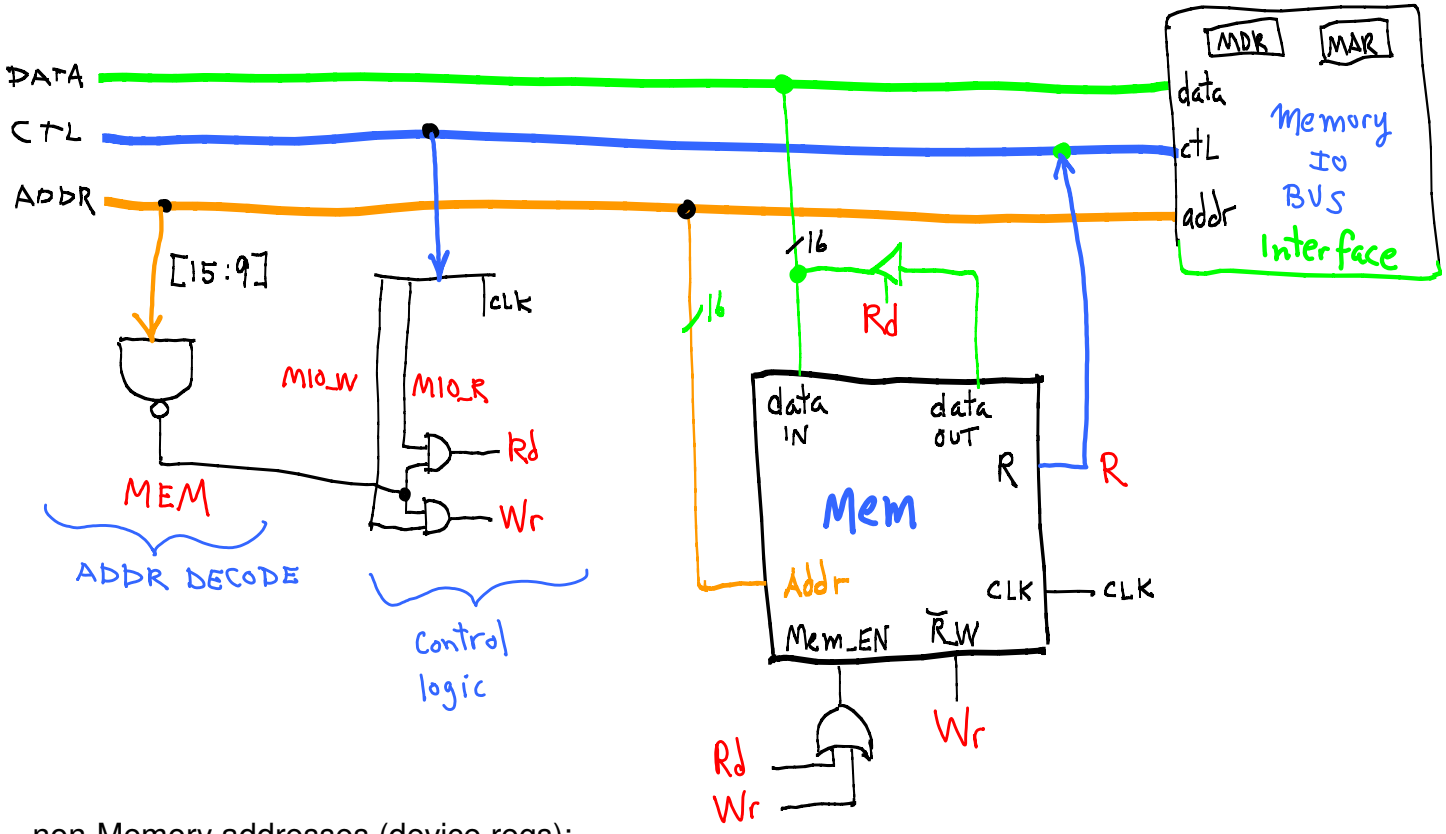
The KBDR circuit is similar.
Address decoders share logic:

KBSR: AND([15:9]) AND NOR([8:2]) AND NOT([1]) AND NOT([0])
KBDR: AND([15:9]) AND NOR([8:2]) AND [1] AND NOT([0])

TRI-STATE, non-Inverting Buffer



Memory, just another device



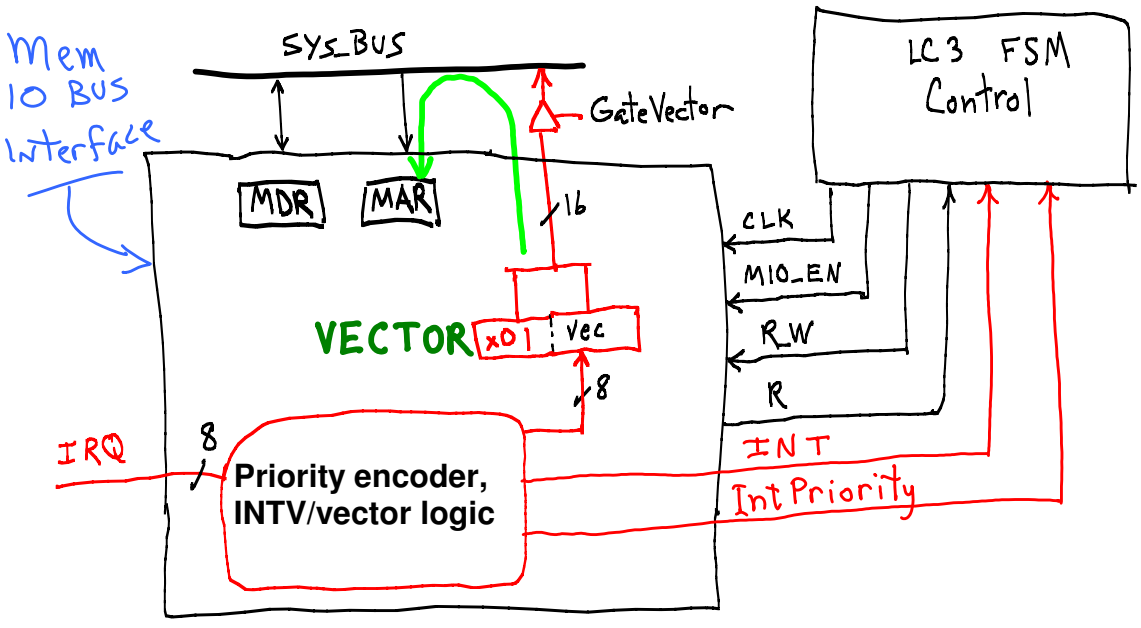
non-Memory addresses (device regs):

F	E	0	0	through	F	F	F	F
1111	1110	0000	0000		1111	1111	1111	1111

ADDR[15:9] all 1s? ==> NOT a Memory address.

Interrupts

Device i sends request on IRQ[i]. Interrupts if higher priority.
Address of interrupt/exception routine's vector is in VECTOR register.
Address of routine (vector) goes to MDR, PC.



Privilege exception:
VECTOR <== x0100

Illegal opcode exception:
VECTOR <== x0101

Keyboard interrupt:
VECTOR <== x0180
IntPriority <== 4

Not an interrupt/exception:
PSR.Priority == 0.

Device with IntPriority == 0:
can't interrupt?

Additional I/O Devices?