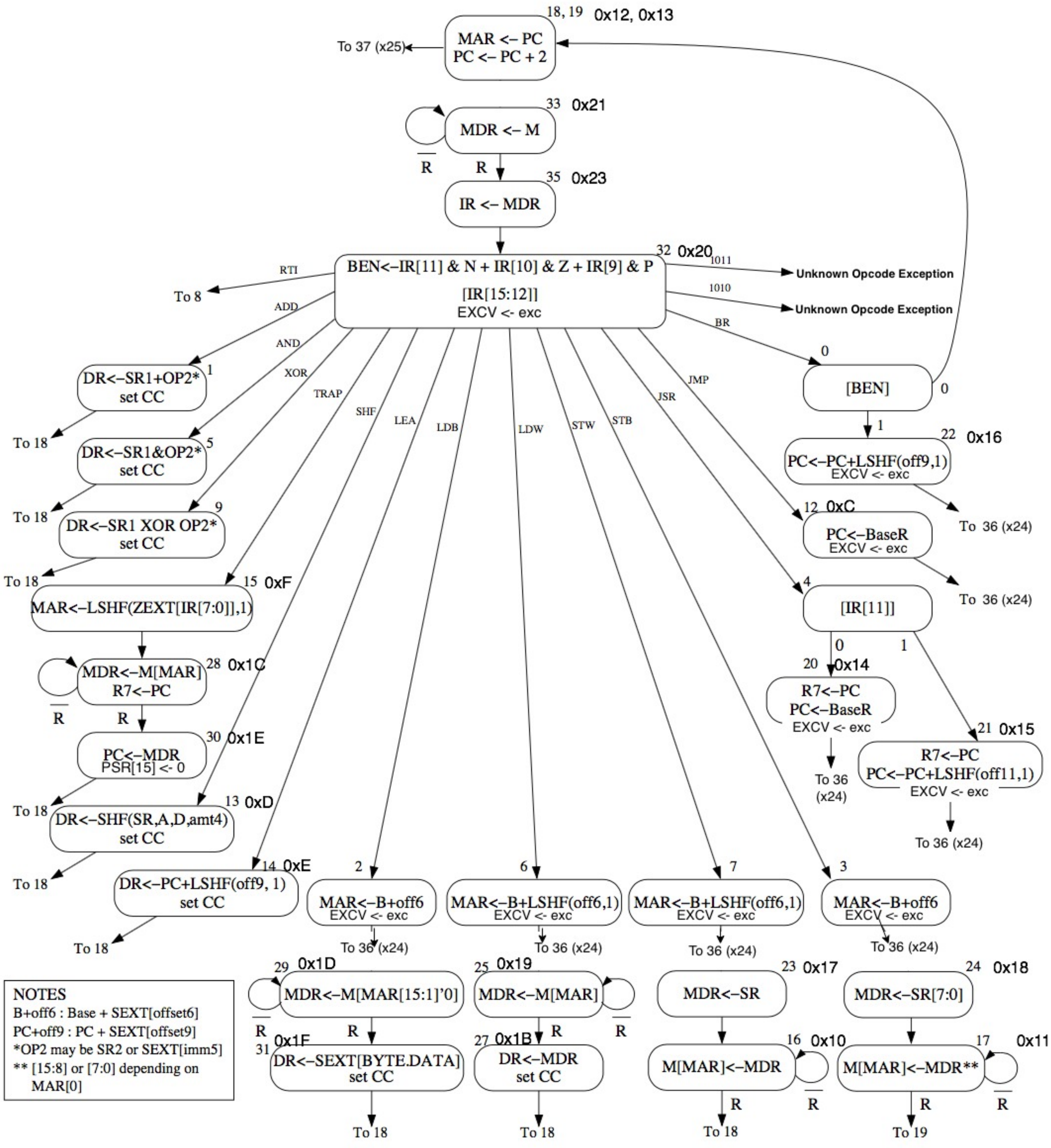
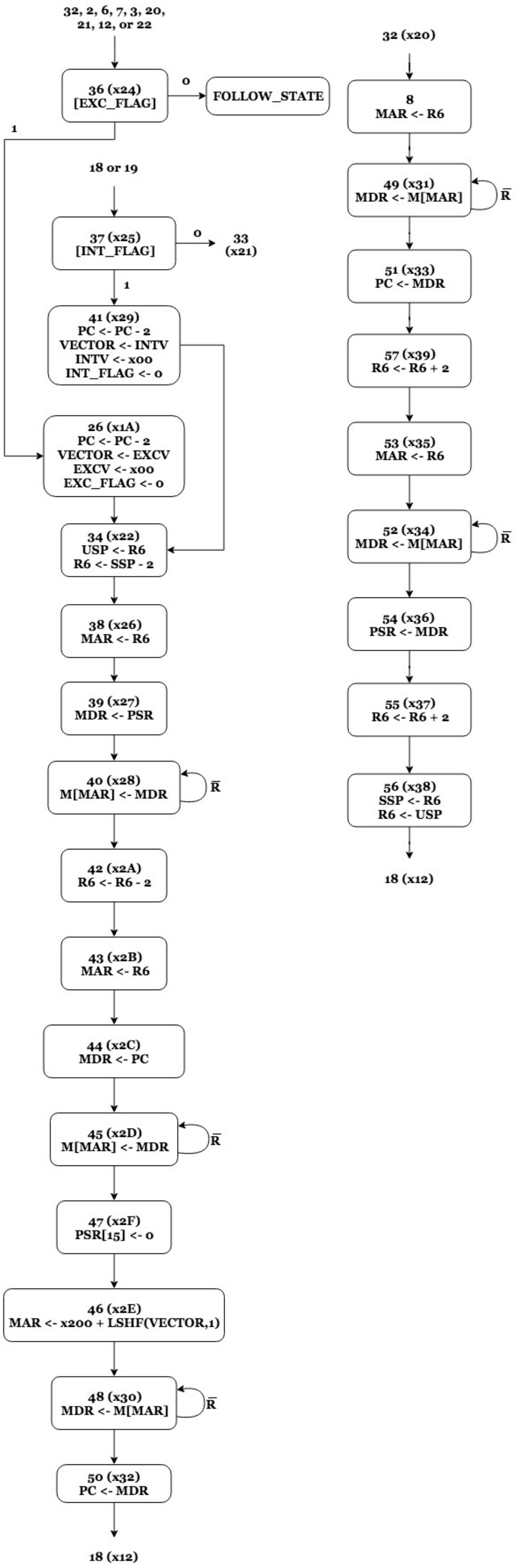
Lab 4 Readme

Michael Coulter
mrc3463



NOTES
B+off6 : Base + SEXT[offset6]
PC+off9 : PC + SEXT[offset9]
*OP2 may be SR2 or SEXT[imm5]
** [15:8] or [7:0] depending on
MAR[0]



Additional Notes

All states check for timer interrupt (INTV <- int) and INT_FLAG.
Any state that loads EXCV will also load EXC_FLAG, FOLLOW_STATE, and will also have state 36 as its next state. However, if no exception is detected in state 36, state 36's next state will be the state that logically follows the state that loaded EXCV (the state's J bits), denoted as FOLLOW_STATE.
FOLLOW_STATE can be 29, 25, 23, 24, 18, or any of the opcode states (8, 1, 5, etc.)
"int" represents the interrupt vector (eg. x00 or x01).
"exc" represents the exception vector (eg. x00, x02, x03, or x04).

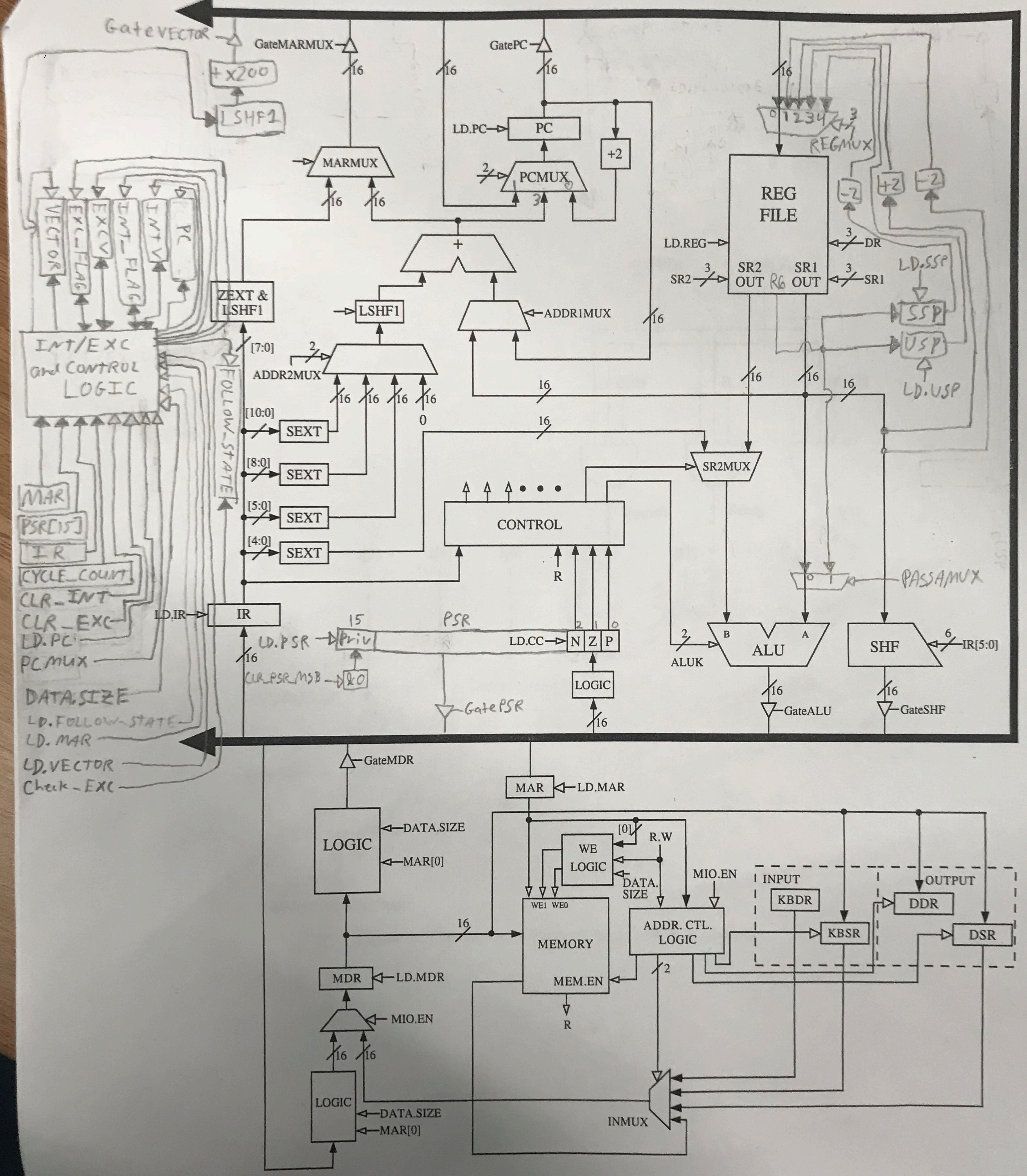


Figure 3: The LC-3b data path

New Datapath Structures

- Extended drawing of PSR register to include privilege bit and supporting logic
- PASSAMUX and supporting logic
- SSP register and supporting logic
- USP register and supporting logic
- REGMUX and supporting logic
- VECTOR register and supporting logic
- Large piece of logic on the left is the extension of the control store and “CONTROL” logic. Uses the listed registers and signals to determine the values of the PC, INTV, INT_FLAG, EXCV, EXC_FLAG, VECTOR, and FOLLOW_STATE registers at the appropriate times.

New Control Signals

LD_PSR - allows the PSR register to be loaded from the BUS
CLR_PSR_MSB - will clear bit 15 of the PSR
LD_USP - allows the USP register to be loaded from R6
LD_SSP - allows the SSP register to be loaded from R6
LD_VECTOR - allows the VECTOR register to be loaded from INTV or EXCV
LD_FOLLOW_STATE - allows the FOLLOW_STATE register to be loaded from state's J bits
CLR_INT - sets the INTV and INT_FLAG registers to 0
CLR_EXC - sets the EXCV and EXC_FLAG registers to 0
REGMUX - three bit select signal that loads a specified register in the register file with following cases:

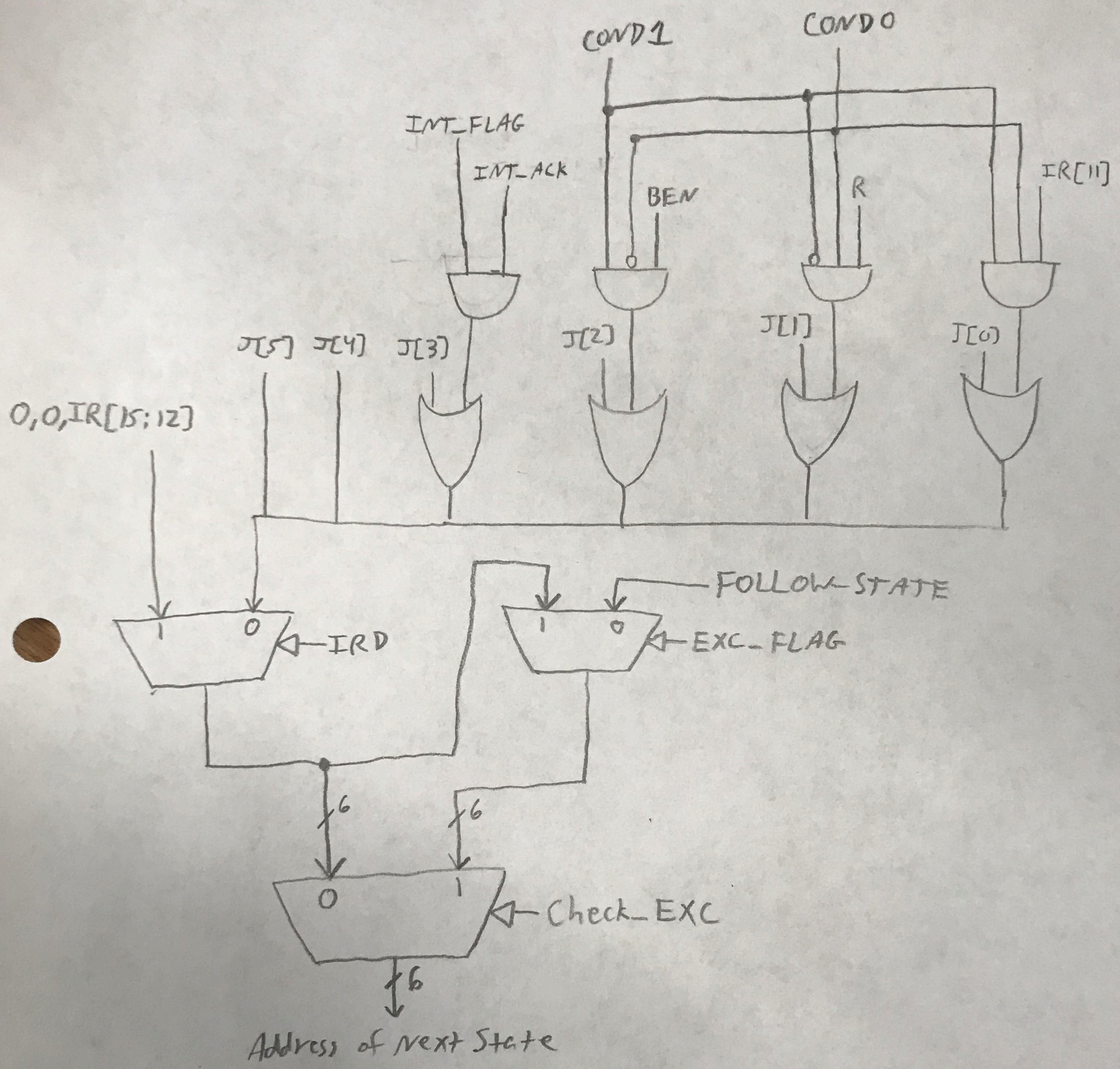
0 - BUS
1 - (R6 - 2)
2 - (R6 + 2)
3 - USP
4 - (SSP - 2)

INT_ACK - similar to the COND bits, will determine the next state for state 37.

If INT_FLAG == 0, next state = 33 (x21).
If INT_FLAG == 1, next state = 41 (x29).
(see microsequencer).

GateVECTOR - allows (x200 + LSHF(VECTOR,1)) to be gated onto the BUS
GatePSR - allows the PSR register to be gated onto the BUS
Check_EXC - for state 36, selects between MUX(J bits, 0,0,IR[15:12]) and MUX(FOLLOW_STATE, MUX(J bits, 0,0,IR[15:12])). (see microsequencer).
PASSAMUX - one bit select signal that sets ALU value A with following cases:

0 - SR1 register
1 - R6


COND1
COND0
INT_FLAG
INT_ACK
BEN
R
IR[11]
J[5]
J[4]
J[3]
J[2]
J[1]
J[0]
0,0,IR[15:12]
1
0
IRD
FOLLOW_STATE
1
0
EXC_FLAG
6
6
0
1
Check_EXC
6
Address of Next State

Microsequencer Changes

- The AND gate with INT_FLAG and INT_ACK ORed with J[3] allow the interrupt check state 37 to go to state 33 if no interrupt is detected, and state 41 if there is an interrupt detected.
- The two additional MUXes allow the exception check state 36 to go to state 26 if an exception is detected, and to the FOLLOW_STATE if no exception is detected.

State	FOLLOW_STATE
32	opcode
2	29
6	25
7	23
3	24
20	18
21	18
12	18
22	18