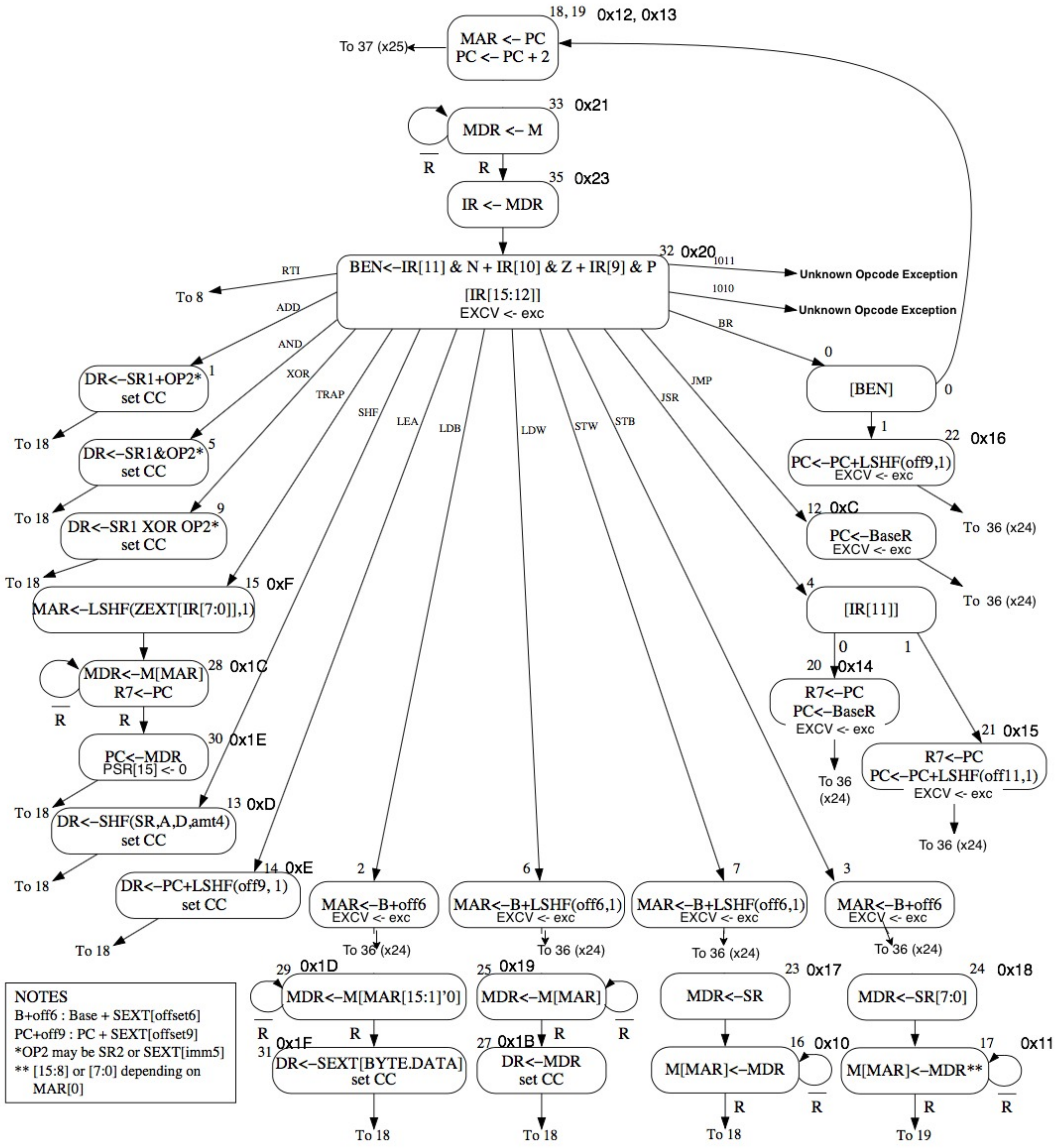
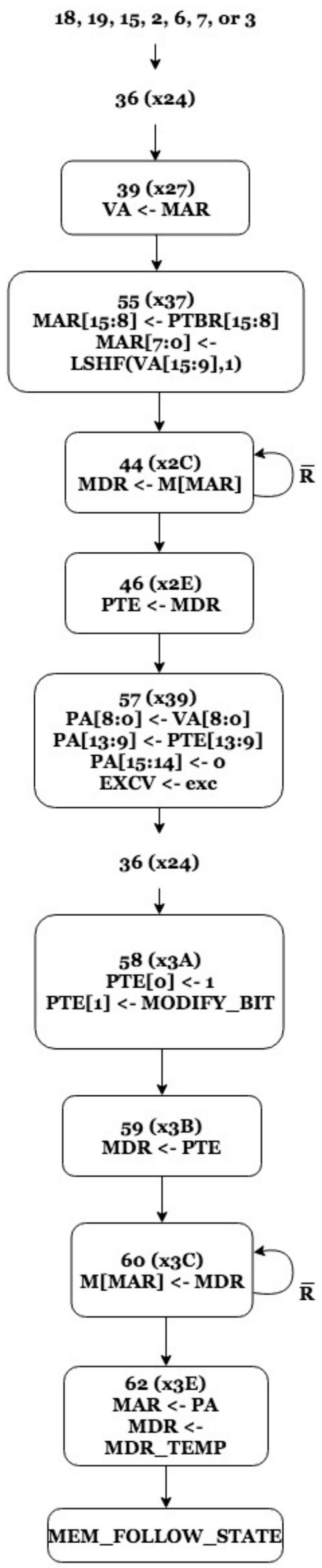
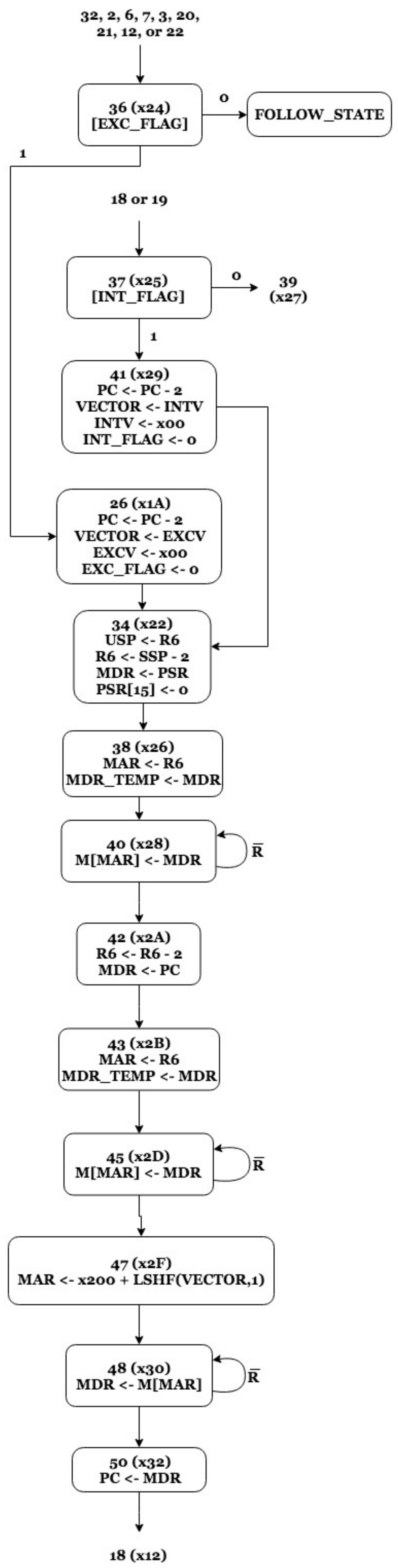


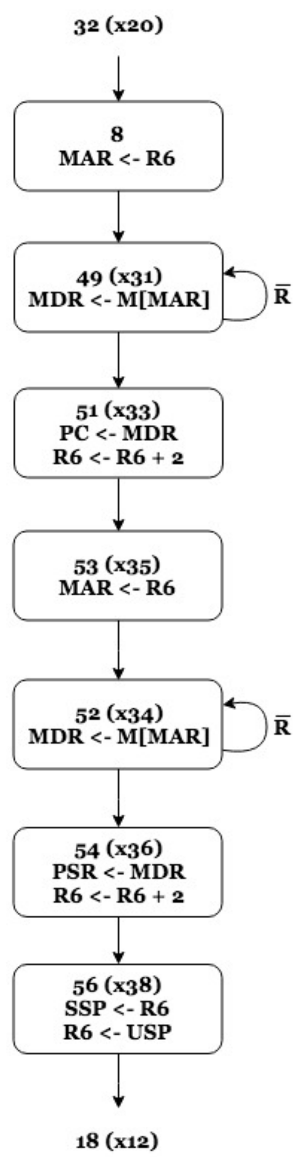
NOTES
B+off6 : Base + SEXT[offset6]
PC+off9 : PC + SEXT[offset9]
*OP2 may be SR2 or SEXT[imm5]
** [15:8] or [7:0] depending on MAR[0]

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).







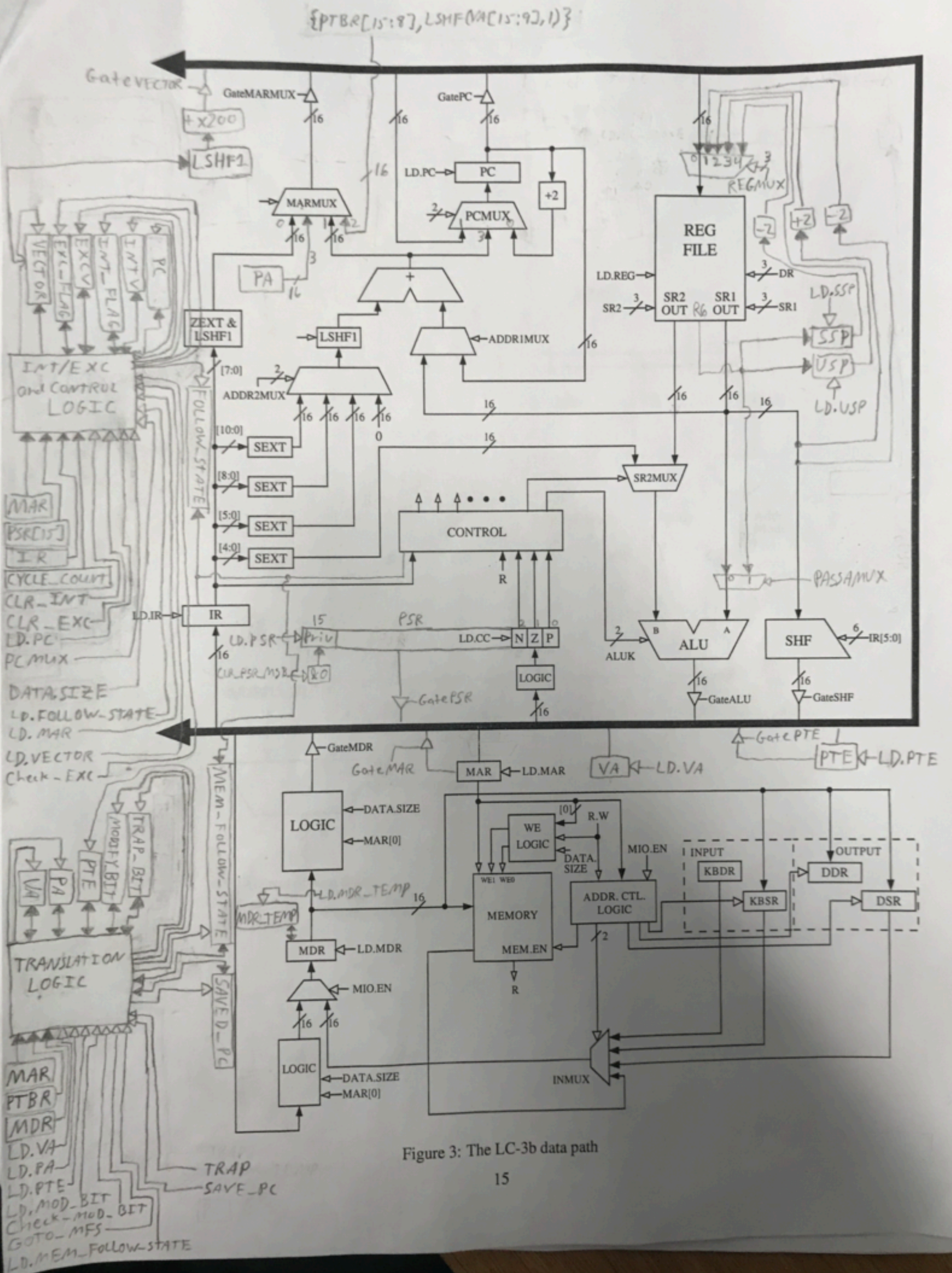
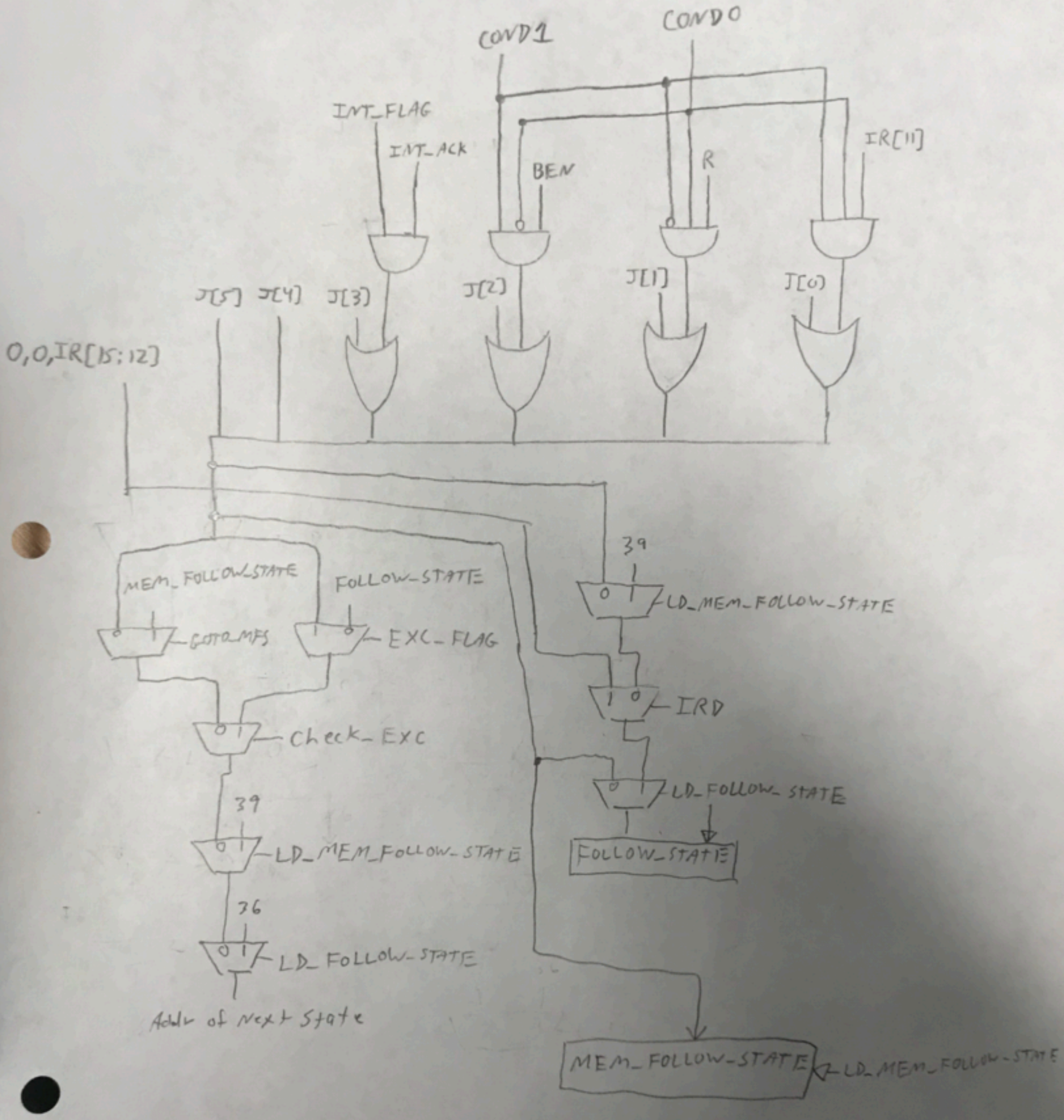


Figure 3: The LC-3b data path

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.
- MDR_TEMP register and supporting logic
- GateMAR logic
- VA register and supporting logic
- PTE register and supporting logic
- additional MARMUX logic
- Large piece of logic on the bottom left called “TRANSLATION LOGIC” is an extension of the control store, “CONTROL” logic block, and the “INT/EXC and CONTROL LOGIC” block. Uses the listed registers and signals to determine the values of the VA, PA, PTE, MODIFY_BIT, TRAP_BIT, SAVED_PC, and MEM_FOLLOW_STATE registers at the appropriate times.


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]
39
0 1
LD_MEM_FOLLOW_STATE
MEM_FOLLOW_STATE
FOLLOW_STATE
0 1
GOTO_MFS
1 0
EXC_FLAG
1 0
IRD
0 1
Check_Exc
39
0 1
LD_MEM_FOLLOW_STATE
0 1
LD_FOLLOW_STATE
FOLLOW_STATE
36
0 1
LD_FOLLOW_STATE
Addr of Next State
MEM_FOLLOW-STATE
LD_MEM_FOLLOW-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 MEM_FOLLOW_STATE register is used for states returning from the translation process.
- Several MUXes are used to properly set the next state, and the FOLLOW_STATE and MEM_FOLLOW_STATE registers.

State	NEXT_STATE	FOLLOW_STATE	MEM_FOLLOW_STATE
32	36	opcode	NA
37	39	NA	33
15	39	NA	28
2	36	39	29
6	36	39	25
7	36	39	23
3	36	39	24
20	36	18	NA
21	36	18	NA
12	36	18	NA
22	36	18	NA
38	39	NA	40
43	39	NA	45
47	39	NA	48
8	39	NA	49
53	39	NA	52
57	36	58	NA

Control Signals for Exceptions and Interrupts

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

Control Signals for Virtual Memory

LD_VA - allows the VA register to be loaded from the BUS
LD_PA - allows the PA register to be loaded like so:

PA[8:0] <— VA[8:0]
PA[13:9] <— PTE[13:9]
PA[15:14] <— 0

Since the PA is formed in the state with this control signal set, LD_PA indicates that checks for protection and page fault exceptions need to be carried out as well.

LD_PTE - allows the PTE register to be loaded. PTE is either loaded with the MDR, or bits 1 and 0 are set depending on the Check_MOD_BIT signal.
LD_MOD_BIT - if set to 1, sets the MODIFY_BIT register to 1
GateMAR - allows the MAR register to be gated onto the BUS
Check_MOD_BIT - determines what will be latched into the PTE, MODIFY_BIT, and TRAP_BIT registers with following cases:

0 - PTE <— MDR
1 - PTE[0] <— 1
PTE[1] <— MODIFY_BIT
MODIFY_BIT <— 0
TRAP_BIT <— 0

GatePTE - allows the PTE register to be gated onto the BUS
LD_MEM_FOLLOW_STATE - allows the MEM_FOLLOW_STATE register to be loaded from state's J bits
GOTO_MFS - if set to 1, the next state will be the current value of the MEM_FOLLOW_STATE register. It also indicates that the MDR needs to be loaded from the MDR_TEMP register.
TRAP - if set to 1, indicates that the current instruction being processed is a TRAP instruction, and sets the TRAP_BIT register to 1.
LD_MDR_TEMP - allows the MDR_TEMP register to be loaded from the MDR
SAVE_PC - allows the SAVED_PC register to be loaded from the PC
MARMUX1, MARMUX0 - 2-bit select signal that will select values to be gated onto the BUS and eventually loaded into the MAR with the following cases:

0 - LSHF(ZEXT[IR[7:0]],1)
1 - ADDER
2 - {PTBR[15:8],LSHF(VA[15:9],1)}
3 - PA