SRC
IN/OUT RTN

If we modify the 1-bus SRC architecture to have a separate I/O address space (denoted by IO[] instead of MEM[]), we can use IN and OUT instructions to read from and write to I/O ports instead of having to use memory-mapped I/O. If we use shared address and data lines, we can do this by adding a single new control signal, IO.H. This new signal is asserted along with READ.H during the execution of an IN instruction and along with WRITE.H during the execution of an OUT instruction to distinguish I/O reads and writes from memory reads and writes.

If the IN and OUT instructions use the same format as the LD and ST instructions (with full displacement-based addressing), the abstract RTN (fragment) is:

IN (:=op=7) → R[ra] ← IO[disp] ;

OUT (:=op=25) → IO[disp] ← R[ra] ;

IN Instruction Concrete RTN and Control Sequence

STEP	RTN	Control Sequence
T0	MA <- PC : C <- PC + 4 ;	PCout,MAin,INC4,Cin
T1	MD <- M[MA] : PC <- C ;	Cout,PCin,MDrd,Read,Wait
T2	IR <- MD ;	MDout,IRin
T3	A <- ((rb=0) -> 0 : (rb /=0) -> R[rb]) ;	Grb,BAout,Ain
T4	C <- A + c2{sign-extended} ;	c2out,ADD,Cin
T5	MA <- C ;	Cout,MAin
T6	MD <- IO[MA] ;	IO,MDrd,Read,Wait
T7	R[ra] <- MD ;	MDout,Gra,Rin,End

OUT Instruction Concrete RTN and Control Sequence

STEP	RTN	Control Sequence
T0	MA <- PC : C <- PC + 4 ;	PCout,MAin,INC4,Cin
T1	MD <- M[MA] : PC <- C ;	Cout,PCin,MDrd,Read,Wait
T2	IR <- MD ;	MDout,IRin
T3	A <- ((rb=0) -> 0 : (rb /=0) -> R[rb]) ;	Grb,BAout,Ain
T4	C <- A + c2{sign-extended} ;	c2out,ADD,Cin
T5	MA <- C ;	Cout,MAin
T6	MD <- R[ra] ;	Gra,Rout,MDbus
T7	IO[MA] <- MD ;	IO,MDwr,Write,Wait,End