**Legend**

HLT – Halt Clock

MPI – Memory Pointer In

MI – Memory Data In

MO – Memory Data Out

CI – Program Counter In

CO – Program Counter Out

CC – Program Counter Count

II – Instruction Register In

IO – Instruction Register Out

GPI – General Register Pointers In

GPO – General Register Destination Pointer Override

GI – General Register Bank In

GO – General Register Bank Out

SC – Stack Pointer Count

SR – Stack Pointer Reverse

SI – Stack Data In

SO – Stack Data Out

RC – Return Pointer Count

RR – Return Pointer Reverse

RI – Return Data In

RO – Return Data Out

AO – ALU Result Out

ASU – ALU Subtract

AIN – ALU Increment

ADC – ALU Decrement

ANT – ALU Not

AND – ALU And

AOR – ALU Or

AXR – ALU Xor

ASH – ALU Shift Left

ART – ALU Rotate Left

FI – Flags Register In

FO – Flags Register Out

TAI – Temporary Register Address Bus In

TAO – Temporary Register Address Bus Out

TDI – Temporary Register Data Bus In

TDO – Temporary Register Data Bus Out

**Fetch**

CO, MPI, MO, II

CC, [else]

**Mov**

CO, MPI, MO, II (Fetch instruction)

CC, IO, GPI (Store Reg-Sel)

GO, TDI (Store value to Temp)

CO, MPI, MO, GPI (Fetch and store Reg-Sel)

CC, TDO, GI (Copy value to Reg)

**Wait**

[null] (Do nothing; still respond to interrupts)

**Lda**

CO, MPI, MO, II (Fetch instruction)

CC, IO, GPI (Store Reg-Sel)

CO, MPI, MO, TAI (Store address in Temp)

CC, TAO, MPI (Fetch data at address)

Source is written to Temp register. Dest is overridden to the output of the Regs. Temp is added to Dest and written to Dest.

For ADD… GO and GPO and TI controls allow Source Pointer to place Source in Temp Register; then GI and TO allows Source to be added to Destination.

ALU always uses Source Register (which can be overwritten by Destination Pointer to be Destination Register) for Operand A and Temporary Register as Operand B.