RTL descriptions for the Assembly Language instructions of the ToyPacer processor.

Clock cycles 0, 1 and 2 will be the same for all instructions. The actions to be performed during these clock cycles are listed below:

0. If IEB and IRQ $\neq 0$ then
   IPC = PC; PC = Port[$IRQ_{1:0}$]; IEB = 0;
1. IR = Mem[PC]; PC = PC + 2;
2. MAR = $DSR_{5:0}||IR_{11:2}$; A = Reg[$IR_{1:0}$] ; B = Reg[$IR_{3:2}$] ;
   ALUOut = PC + SE($IR_{11:2}$<<1);

Load:
3. MDR = Mem[MAR]
4. Reg[$IR_{1:0}$] = MDR

Store:
3. Mem[MAR] = A

Move:
3. Reg[$IR_{3:2}$] = A

Add:
3. ALUOut = A + B
4. Reg[$IR_{3:2}$] = ALUOut

ADDI:
3. ALUOut = A + ZE($IR_{11:2}$)
4. Reg[$IR_{3:2}$] = ALUOut

SUB:
3. ALUOut = A - B
4. Reg[$IR_{3:2}$] = ALUOut

SUBI:
3. ALUOut = A - ZE($IR_{11:2}$)
4. Reg[$IR_{3:2}$] = ALUOut

BGE:
3. If A-B ≥ 0 then PC = ALUOut

RFI:
3. IEB = 1
4. PC = IPC

ASSERT:
3. Port[$IR_{1:0}$] = $IR_{11:2}$

SETSEG
3. DSR = $IR_{7:2}$

MASKI:

3. IEB = NOT(IEB)

Components list (identified using the RTL descriptions)

Components	Inputs	Outputs	Control signals (Each signal is 1 bit wide)
Mem	$MemAddr_{15:0,}$ $MemDI_{15:0}$	$MemDO_{15:0}$	MemRead, MemWrite
RegFile	$RegAddr1_{1:0}$, $RegAddr2_{1:0}$, $RegDI_{15:0}$	$RegDO1_{15:0}$, $RegDO2_{15:0}$	PortWrite
Port	$PortAddr_{1:0}$, $PortDI_{15:0}$	$PortDO_{15:0}$	PortWrite
PC	$PCIn_{15:0}$	$PCOut_{15:0}$	PCWrite
IR	$IRIn_{15:0}$	$IROut_{15:0}$	IRWrite
DSR	$DSRIn_{5:0}$	$DSROut_{5:0}$	DSRWrite
MAR	$MARDI_{15:0}$	$MARDO_{15:0}$	MARWrite
MDR	$MDRDI_{15:0}$	$MDRDO_{15:0}$	MDRWrite
A	$ADI_{15:0}$	$ADO_{15:0}$	-
B	$BDI_{15:0}$	$BDO_{15:0}$	-
ALUOut	$ALUOutDI_{15:0}$	$ALUOutDO_{15:0}$	-
IEB		$IEBDO_{0:0}$	IEBSet,IEBClr
IPC	$IPCDI_{15:0}$	$IPCDO_{15:0}$	IPCWrite
IRQ		$IRQDO_{1:0}$	
ALU	$ALUDI1_{15:0}$,$ALUDI2_{15:0}$	$ALUDO_{15:0}$	$ALUOP_{0:0}$ (0=> ADD; 1 => SUB)
ZE	$ZEDI_{9:0}$	$ZEDI_{15:0}$	-
SE	$SEDI_{9:0}$	$SEDO_{15:0}$	-

Multiplexers

PCMux : 0 => ALUDO ; 1 => PortDO ; 2 => IPCDO
MemMux: 0 => PCDO ; 2 => MARDO
RegMux: 0 => MDRDO ; 1 => ADO ; 2 => ALUOutDO
ALUMux1 : 0 => PC ; 1 => ADO
ALUMux2 : 0 => 2; 1 => SE ($IR_{11:2}$||0); 2 => B; 3 => ZE($IR_{11:2}$)