Code:

```
Subdesign Toro680
  (
  ClkM27, Reset_b, ClkSel[2..0]                         :Input;
  LdMAR, Rd_b, Wr_b, ClkDisp ,MemClkO                   :Output;
  StOut[5..0],BusOut[7..0]                              :Output;
  Bus[7..0]                                             :BIDIR;
  )
Variable
  Ctr[27..0], MemClk, CpuClk                            :Dff;
  OeAClk, OeAdelayed                                    :Dff;
  PC[7..0], IR[7..0], AC[7..0], C                       :Dff;
  PCt[7..0], At[7..0]                                   :Tri;
  IncPC, LdPC, OePC, LdIR, LdA, OeA                     :Node;
          Lw,_And , _Or , Add , _Xor , shift            :Node;
          inh , imm , dir , ind                         :Node;
          ALU[8..0] ,aluop, sta , br                    :Node;
          BCT                                           :Node;
  Sum[8..0], N, Z                                       :Node;
          Reset                                         :Node;
          write , read                                  :Node;
  Busnode[7..0]                                         :Tri_State_Node;
          s[2..0]                                       :Dff;
          I0,I1,Ad0,Ad1,Ad2,Ex                          :Node;
--   State                                              :Machine
--    With States(I0,I1,Ad0,Ad1,Ad2,Ex);  % Rename to S0 to S5 if you prefer %

Begin
  Ctr[].clk = ClkM27;
  Ctr[].d = Ctr[].q+1;
  MemClk.clk = ClkM27;
  MemClk.d = (ClkSel[]==0)&Ctr[27] # (ClkSel[]==1)&Ctr[26]
        # (ClkSel[]==2)&Ctr[24] # (ClkSel[]==3)&Ctr[22]
        # (ClkSel[]==4)&Ctr[20] # (ClkSel[]==5)&Ctr[18]
        # (ClkSel[]==6)&Ctr[16] # (ClkSel[]==7)&Ctr[6];
  MemClkO = MemClk.q;  % Clock being sent to Memory board %
  OeAClk.clk = ClkM27;
  OeAClk.d = (ClkSel[]==0)&Ctr[25] # (ClkSel[]==1)&Ctr[24]
        # (ClkSel[]==2)&Ctr[22] # (ClkSel[]==3)&Ctr[20]
        # (ClkSel[]==4)&Ctr[18] # (ClkSel[]==5)&Ctr[16]
        # (ClkSel[]==6)&Ctr[14] # (ClkSel[]==7)&Ctr[4];
  OeAdelayed.clk = OeAClk.q; % OeAClk is 4 times faster than MemClk %
  OeAdelayed.d = OeA;       % OeAdelayed.q is 1/4 MemClk cycle behind OeA %
  At[].oe = OeAdelayed.q;   % The RAM chip requires significant hold time %
  PCt[].oe = OePC;
```

```
CpuClk.clk = ClkM27;
CpuClk.d = MemClk.q;  % CpuClk is 1/27M sec (37 ns) behind MemClk %
PC[].clk = CpuClk;
IR[].clk = CpuClk;
AC[].clk = CpuClk;
C.clk = CpuClk;
          s[].clk = CpuClk;
--State.clk = CpuClk;
ClkDisp = CpuClk;     % Clock being displayed on DE2 used by CPU Dffs %

Sum[] = (0,AC[]) # (0,Busnode[]); % extend add to 9 bits for access to Carry %
          BusOut[] = Bus[];

          PCt[].in = PC[].q;
          At[].in = AC[].q;

          Busnode[] = Bus[];
          Busnode[] = PCt[].out ;
          Busnode[] = At[].out;
          Bus[] = Busnode[];

          Reset = !Reset_b;
          Wr_b = !write;
          Rd_b = !read;

          aluop = !IR7 &!IR6;
          sta = IR7 &!IR6;
          br  = !IR7 & IR6;
          inh = !IR5 & !IR4;
          imm = !IR5 & IR4;
          dir = IR5 & !IR4;
          ind = IR5 & IR4;

          I0 = s[2..0].q == 0;
          I1 = s[2..0].q == 1;
          Ad0 = s[2..0].q == 2;
          Ad1 = s[2..0].q == 3;
          Ad2 = s[2..0].q == 4;
          Ex = s[2..0].q == 5;
          StOut[0..5] = (I0 , I1 , Ad0 , Ad1 , Ad2 , Ex) ;


          Lw =   !IR2 & !IR1 & !IR0 ;
          Add =  !IR2 & !IR1 & IR0 ;
          _And = !IR2 & IR1 & !IR0 ;
          _Or =  !IR2 & IR1 & IR0 ;
          _Xor =  IR2 & !IR1 & !IR0 ;
          shift = IR2 & !IR1 & IR0;
```

```
        s2.d = !Reset & Ad0 & dir # !Reset & Ad0 & imm
                        # !Reset & Ad1 # !Reset & Ad2;
        s1.d = !Reset & I1 # !Reset & Ad0 & ind;
        s0.d = !Reset & I0 # !Reset & Ad0 & inh
                        # !Reset & Ad0 & imm # !Reset & Ad0 & ind
                        # !Reset & Ad2;

        BCT    = (!IR2 & !IR1 & !IR0) # (!IR2 & !IR1 & IR0) & C # (!IR2 & IR1 & !IR0) & N # (!IR2 &  IR1 &
IR0) &  Z #
                                (IR2 & !IR1 & IR0) &  !C # (IR2 & IR1 & !IR0) & !N # (IR2 & IR1 & IR0) & !Z
;

        LdPC  = Ex & br & BCT;
        LdIR  = I1;
        LdA   = (Ad0 & inh) # Ex & aluop;
        LdMAR = I0 # Ad0 # Ad1 # Ad2;
        write  = Ex & sta;
        OePC  = I0 # Ad0;
        OeA   = Ex & sta ;
        read   = I1 # Ad1 # Ad2 # (Ex & br & BCT) # aluop & Ex;
        IncPC = I0  # Ad0 ;

        ALU[] =  Lw & (0, Bus[]) #
                                Add & ((0,AC[]) + (0,Bus[])) #
                                _And & (0 , AC[] & Bus[]) #
                                _Or & (0 , AC[] # Bus[]) #
                                _Xor & (0 , Bus[] $ AC[]) #
                                shift & (0 , AC7 , AC[7..1]);

        PC[].d = !Reset & !LdPC & !IncPC & PC[].q #
                                !Reset & LdPC & Bus[] #
                                !Reset & IncPC & (PC[] + 1);

        IR[].d = !Reset & !LdIR & IR[].q #
                                !Reset & LdIR & Bus[] ;

        AC[].d = !Reset & !LdA & AC[].q #
                          !Reset & LdA & ALU[7..0] ;

        N = AC7;
        C.d = Ex & aluop & Add & ALU8 # !(Ex & aluop & Add) & C.q;
        Z = AC[] == 0;

End;
```