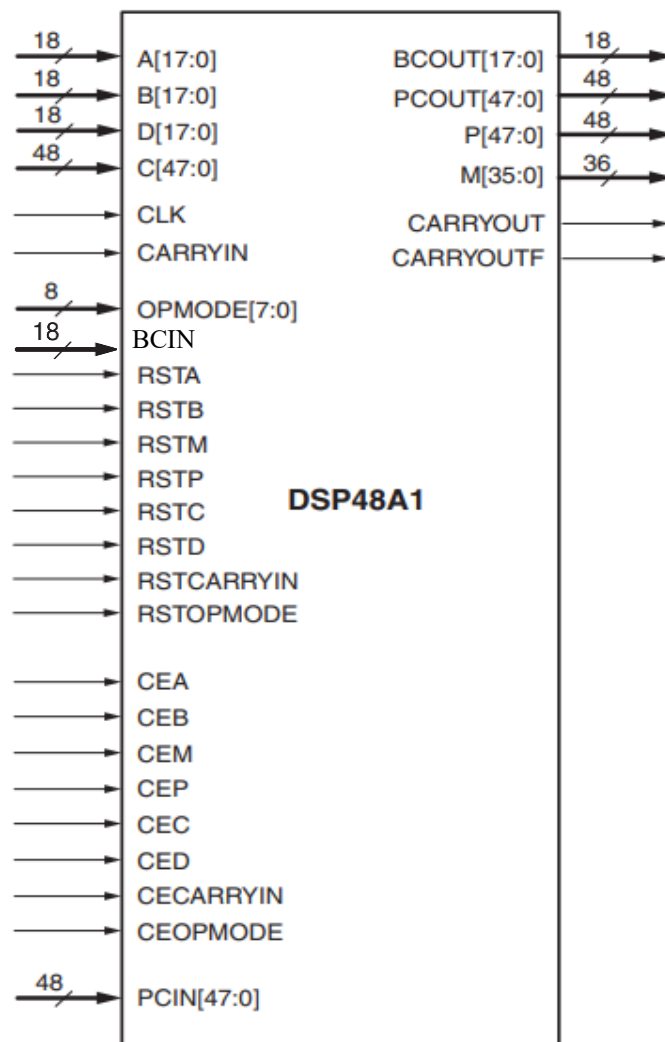




V15 Digital Design Diploma

DSP48A1 Testbench Stimulus



Default Configuration Parameters

First: Ensure that all DSP parameters are set to their default values, as listed below.

```
parameter  A0REG      = 0;
parameter  A1REG      = 1;
parameter  B0REG      = 0;
parameter  B1REG      = 1;
parameter  CREG       = 1;
parameter  DREG       = 1;
parameter  MREG       = 1;
parameter  PREG       = 1;
parameter  CARRYINREG = 1;
parameter  CARRYOUTREG = 1;
parameter  OPMODEREG  = 1;
parameter  CARRYINSEL = "OPMODE5";
parameter  B_INPUT    = "DIRECT";
parameter  RSTTYPE    = "SYNC";
```

Test Reset Operation

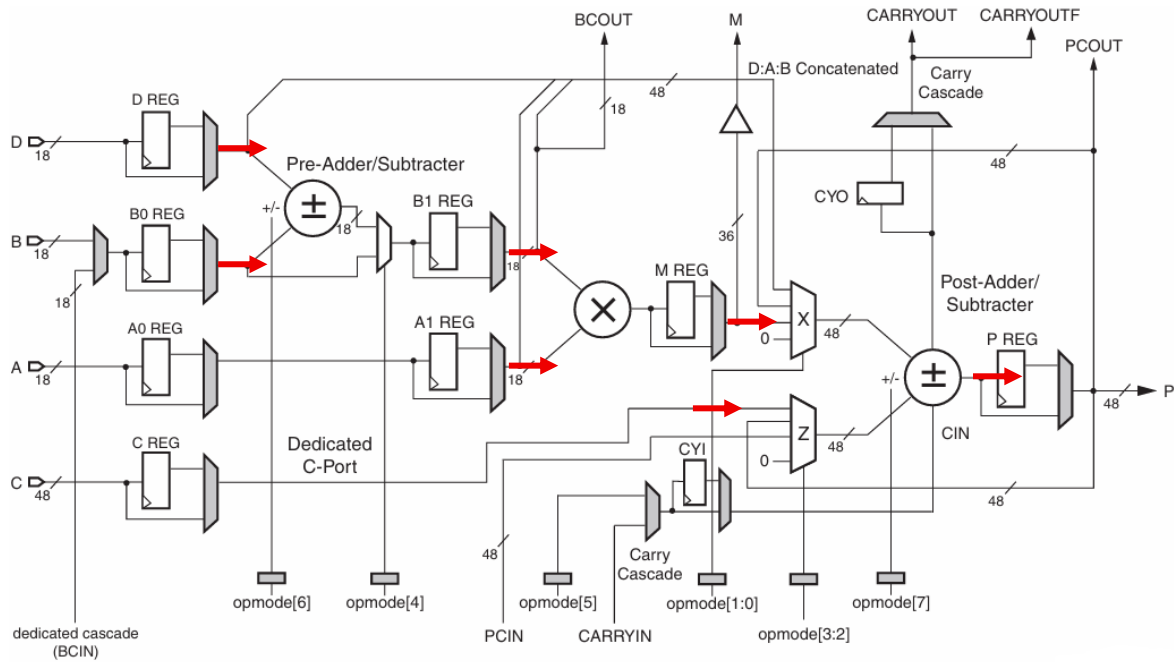
Second: Test the reset operation by setting all active-high reset signals to 1. Then, deactivate the reset and enable the clock signals to verify the functionality of all DSP paths.

```
RSTA = 1;    RSTB = 1;    RSTM      = 1;    RSTP      = 1;
RSTC = 1;    RSTD = 1;    RSTCARRYIN = 1;    RSTOPMODE = 1;
A    = $random; B      = $random; D      = $random; C      = $random;
PCIN = $random; OPMODE = $random; CARRYIN = $random; BCIN   = $random;
CEA  = $random; CEB    = $random; CEC    = $random; CEM     = $random;
CEP  = $random; CED    = $random; CECARRYIN = $random; CEOPMODE = $random;

@(negedge CLK);
if ({BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF} != 0) begin
    $display("Error in DSP Reset Operation");
    $stop;
end
else $display("DSP Reset Operation is Correct");

RSTA = 0; RSTB = 0; RSTM      = 0; RSTP      = 1;
RSTC = 0; RSTD = 0; RSTCARRYIN = 0; RSTOPMODE = 1;
CEA  = 1; CEB  = 1; CEC      = 1; CEM      = 1;
CEP  = 1; CED  = 1; CECARRYIN = 1; CEOPMODE = 1;
```

Test Path 1 (Pre-Sub, Post-Sub & Multiplier)

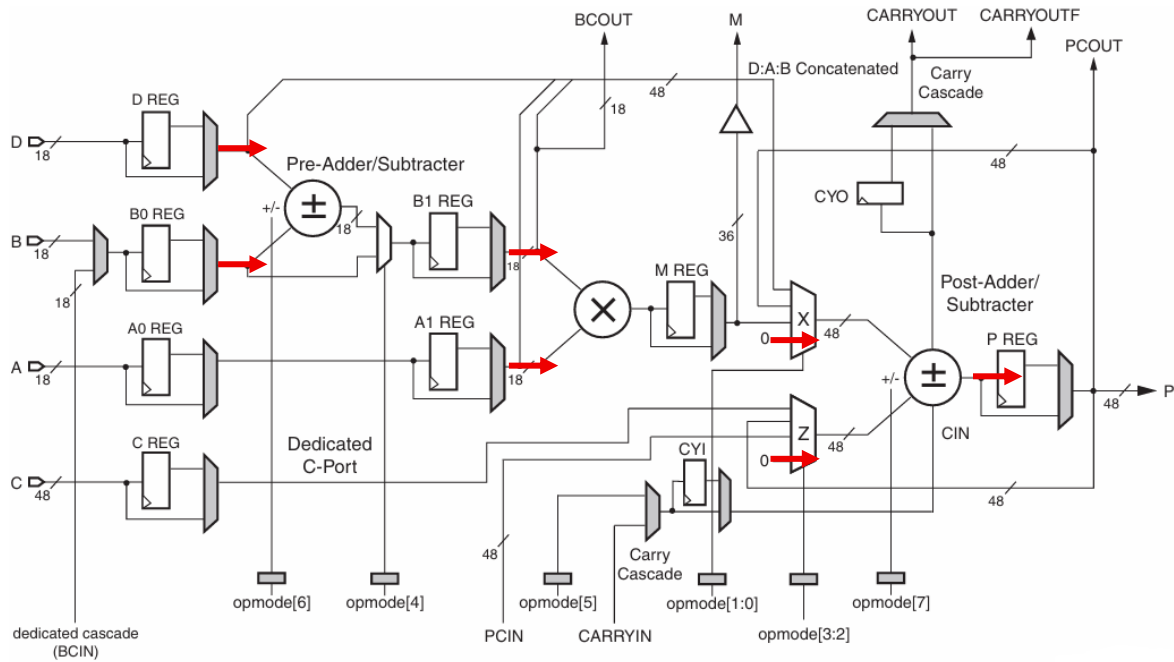


```

OPMODE[7]  = 1;      // Post Subtraction Operation
OPMODE[6]  = 1;      // Pre Subtraction Operation
OPMODE[5]  = 0;      // CARRYIN = OPMODE[5]
OPMODE[4]  = 1;      // Allow Pre out to propagate
OPMODE[3:2] = 2'b11; // Mux(Z) = C-Port
OPMODE[1:0] = 2'b01; // Mux(X) = Multiplier Out
// Input Stimulus
A = 20; B = 10; C = 350; D = 25;
BCIN = $random; PCIN = $random; CARRYIN = $random;
// Calculate Expected Outputs
BCOUT_EXP      = D - B;          // BCOUT = D-B = 'd15 = 'hf
M_EXP          = BCOUT_EXP * A; // M = BCOUT*A = 'd300 = 'h12c
{CARRYOUT_EXP,P_EXP} = C - M_EXP; // P = (C-((D-B)*A)) = 'd50
PCOUT_EXP      = P_EXP;          // PCOUT = P = 'd50 = 'h32
CARRYOUTF_EXP  = CARRYOUT_EXP; // CARRYOUT = CARRYOUTF = 0
// Calculate the Longest Path Delay then make Self Checking
// 4 Flipflops in the longest path (DREG, B1REG, MREG, PREG)
repeat(4) @(negedge CLK);
if ({BCOUT_EXP,PCOUT_EXP,P_EXP,M_EXP,CARRYOUT_EXP,CARRYOUTF_EXP}
    != {BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF}) begin
    $display("Error: DSP Design is Wrong for Path 1");
    $stop;
end
else $display("DSP Path 1 is Correct");

```

Test Path 2 (Pre-Add, Post-Add & Zeros in MUX (X & Z))

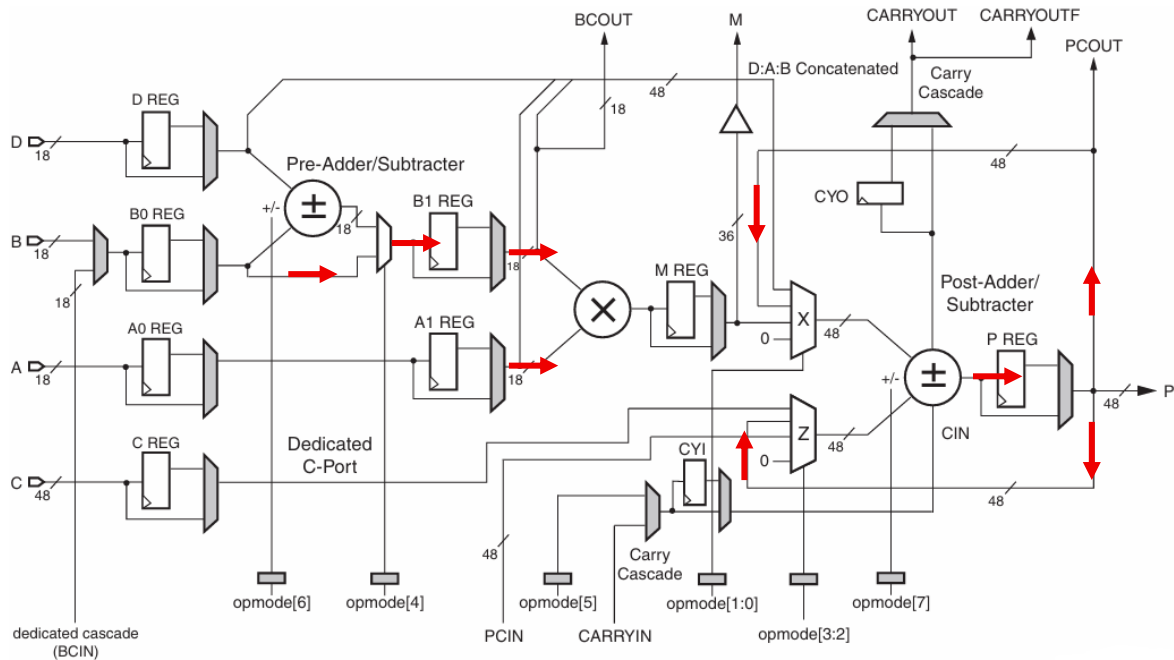


```

OPMODE[7]  = 0;      // Post Addition Operation
OPMODE[6]  = 0;      // Pre Addition Operation
OPMODE[5]  = 0;      // CARRYIN = OPMODE[5]
OPMODE[4]  = 1;      // Allow Pre out to propagate
OPMODE[3:2] = 2'b00; // Mux(Z) = 0
OPMODE[1:0] = 2'b00; // Mux(X) = 0
// Input Stimulus
A = 20; B = 10; C = 350; D = 25;
BCIN = $random; PCIN = $random; CARRYIN = $random;
// Calculate Expected Outputs
BCOUT_EXP      = D + B;          // BCOUT = D+B = 'd35 = 'h23
M_EXP          = BCOUT_EXP * A; // M = BCOUT*A = 'd700 = 'h2bc
{CARRYOUT_EXP,P_EXP} = 0;        // P = 0
PCOUT_EXP      = P_EXP;          // PCOUT = 0
CARRYOUTF_EXP  = CARRYOUT_EXP; // CARRYOUT = CARRYOUTF = 0
// Calculate the Longest Path Delay then make Self Checking
// 3 FlipFlops in the longest path (DREG, B1REG, MREG)
repeat(3) @(negedge CLK);
if ({BCOUT_EXP,PCOUT_EXP,P_EXP,M_EXP,CARRYOUT_EXP,CARRYOUTF_EXP}
    != {BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF}) begin
    $display("Error: DSP Design is Wrong for Path 2");
    $stop;
end
else $display("DSP Path 2 is Correct");

```

Test Path 3 (No Pre, Post-Add & P Feedback)

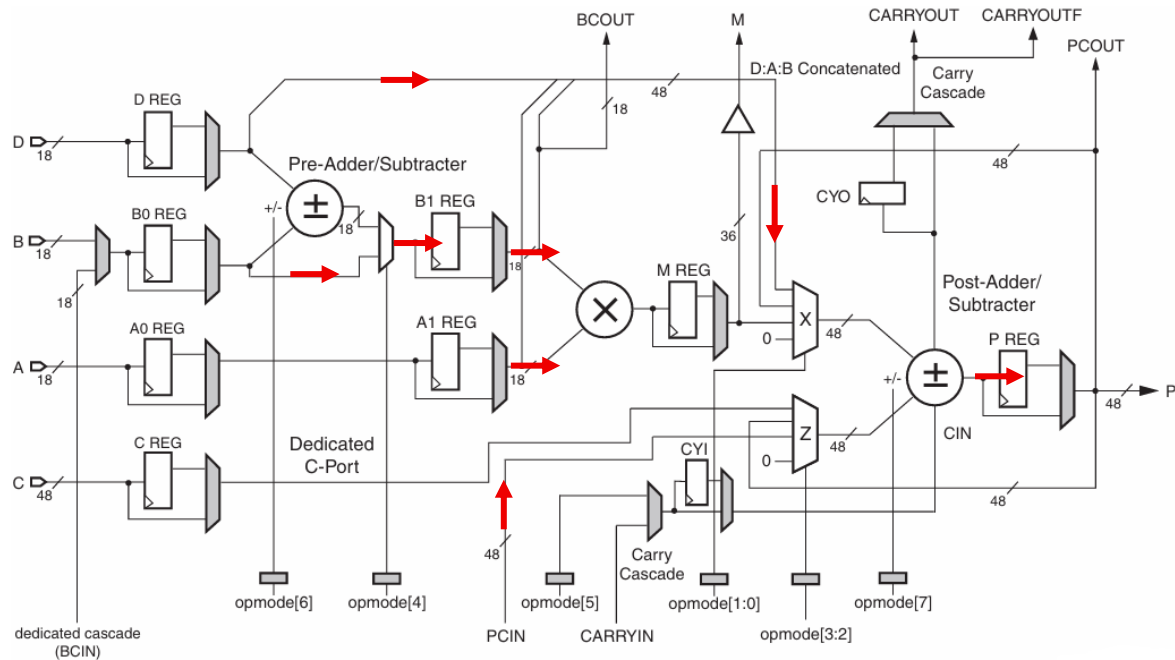


```

OPMODE[7]  = 0;      // Post Addition Operation
OPMODE[6]  = 0;      // Pre Addition Operation
OPMODE[5]  = 0;      // CARRYIN = OPMODE[5]
OPMODE[4]  = 0;      // Don't allow Pre out to propagate
OPMODE[3:2] = 2'b10; // Mux(Z) = P
OPMODE[1:0] = 2'b10; // Mux(X) = P
// Input Stimulus
A = 20; B = 10; C = 350; D = 25;
BCIN = $random; PCIN = $random; CARRYIN = $random;
// Calculate Expected Outputs
BCOUT_EXP      = B;          // BCOUT = B = 'd10  = 'ha
M_EXP          = BCOUT_EXP * A; // M = BCOUT*A = 'd200 = 'hc8
{CARRYOUT_EXP,P_EXP} = {CARRYOUT_EXP,P_EXP};
PCOUT_EXP      = P_EXP;
CARRYOUTF_EXP  = CARRYOUT_EXP;
// Calculate the Longest Path Delay then make Self Checking
// 3 FlipFlops in the longest path (B1REG, MREG, PREG)
repeat(3) @(negedge CLK);
if ({BCOUT_EXP,PCOUT_EXP,P_EXP,M_EXP,CARRYOUT_EXP,CARRYOUTF_EXP}
    != {BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF}) begin
    $display("Error: DSP Design is Wrong for Path 3");
    $stop;
end
else $display("DSP Path 3 is Correct");

```

Test Path 4 (No Pre, Post-Sub, PCIN & Concatenated)



```
OPMODE[7] = 1; // Post Subtraction Operation
OPMODE[6] = 0; // Pre Addition Operation
OPMODE[5] = 1; // CARRYIN = OPMODE[5]
OPMODE[4] = 0; // Don't allow Pre out to propagate
OPMODE[3:2] = 2'b01; // Mux(Z) = PCIN
OPMODE[1:0] = 2'b11; // Mux(X) = D:A:B
// Input Stimulus
A = 5; B = 6; C = 350; D = 25;
BCIN = $random; PCIN = 3000; CARRYIN = $random;
// Calculate Expected Outputs
BCOUT_EXP = B; // BCOUT = B = 'd6 = 'h6
M_EXP = BCOUT_EXP * A; // M = BCOUT*A = 'd30 = 'h1e
{CARRYOUT_EXP,P_EXP} = PCIN - ({D[11:0],A,B} + OPMODE[5]);
PCOUT_EXP = P_EXP; // P = Large Number
CARRYOUTF_EXP = CARRYOUT_EXP; // CARRYOUT = CARRYOUTF = 1
// Calculate the Longest Path Delay then make Self Checking
// 3 FlipFlops in the longest path (B1REG, MREG, PREG)
repeat(3) @(negedge CLK);
if ({BCOUT_EXP,PCOUT_EXP,P_EXP,M_EXP,CARRYOUT_EXP,CARRYOUTF_EXP}
    != {BCOUT,PCOUT,P,M,CARRYOUT,CARRYOUTF}) begin
    $display("Error: DSP Design is Wrong for Path 4");
    $stop;
end
else $display("DSP Path 4 is Correct");
```