Activities (Tuesday, January 21st 2025)

1. Computer A has an overall CPI of 1.3 and can be run at a clock rate of 600MHz. Computer B has a CPI of 2.5 and can be run at a clock rate of 750 Mhz. We have a particular program we wish to run. When compiled for computer A, this program has exactly 100,000 instructions. How many instructions would the program need to have when compiled for Computer B, in order for the two computers to have exactly the same execution time for this program?

$(CPUTime)_A = (Instruction\ count)_A * (CPI)_A * (Clock\ cycle\ Time)_A$

$= (100{,}000) * (1.3) / (600 * 10^6)$ ns

$(CPUTime)_B = (Instruction\ count)_B * (CPI)_B * (Clock\ cycle\ Time)_B$

$= (I)_B * (2.5) / (750 * 10^6)$ ns

Since $(CPUTime)_A = (CPUTime)_B$,

we have to solve for (I)B and get 65000

Or you can solve it that way:

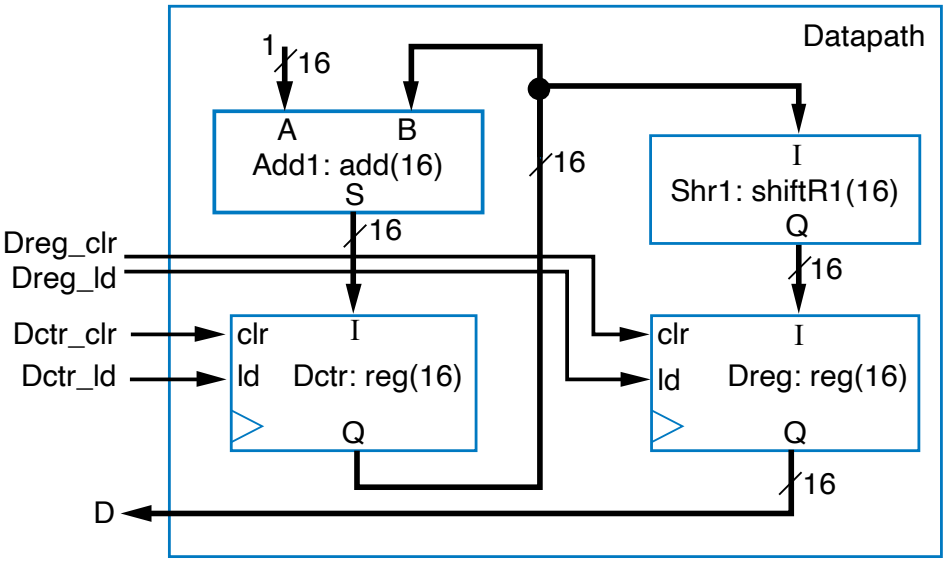
$$\begin{cases} CPI_A = 1.3 \\ Instruction\ Number_A = 10^5 \rightarrow Execution\ time\ for\ A = \dfrac{CPI_A * Instruction\ Number_A}{F_A} = \dfrac{13}{6} * 10^{-4} \\ Ferequency_A = 600\ MHz \end{cases}$$

$$Execution\ Time\ A = Execution\ Time\ B \rightarrow \frac{CPI_B * Instruction\ Number_B}{F_B} = \frac{13}{6} * 10^{-4}$$

$$\rightarrow \frac{2.5 * nstruction\ Number_B}{750 * 10^6} = \frac{13}{6} * 10^{-4} \rightarrow Instruction\ Number_B = 65000$$

1. Draw the block diagram that describes the following Verilog code. You have to show all the inputs, outputs, intermediate signals and blocks. (3 points)

```
module Add16(A, B, S);
   input [15:0] A, B;
   output [15:0] S;
   //details not shown
endmodule

module Reg16(I, Q, clk, clr, ld);
   input [15:0] I;
   input clk, clr, ld;
   output [15:0] Q;
   // details not shown
endmodule

module ShiftR1_16(I, S);
   input [15:0] I;
   output [15:0] S;
   // details not shown
endmodule

module LDM_Datapath(clk, Dreg_clr, Dreg_ld,
                    Dctr_clr, Dctr_ld, D);
   input clk;
   input Dreg_clr, Dreg_ld;
   input Dctr_clr, Dctr_ld;
   output [15:0] D;

   wire [15:0] addC, tempC, shiftC;

   Reg16 Dctr(addC,tempC,clk,Dctr_clr,Dctr_ld);
   Add16 Add1(1, tempC, addC);
   ShiftR1_16 ShiftRight(tempC, shiftC);
   Reg16 Dreg(shiftC,D,clk,Dreg_clr,Dreg_ld);
endmodule
```


Datapath
1
16
A
B
Add1: add(16)
S
16
16
I
Shr1: shiftR1(16)
Q
16
Dreg_clr
Dreg_ld
Dctr_clr
Dctr_ld
clr
ld
I
Dctr: reg(16)
Q
clr
ld
I
Dreg: reg(16)
Q
16
D