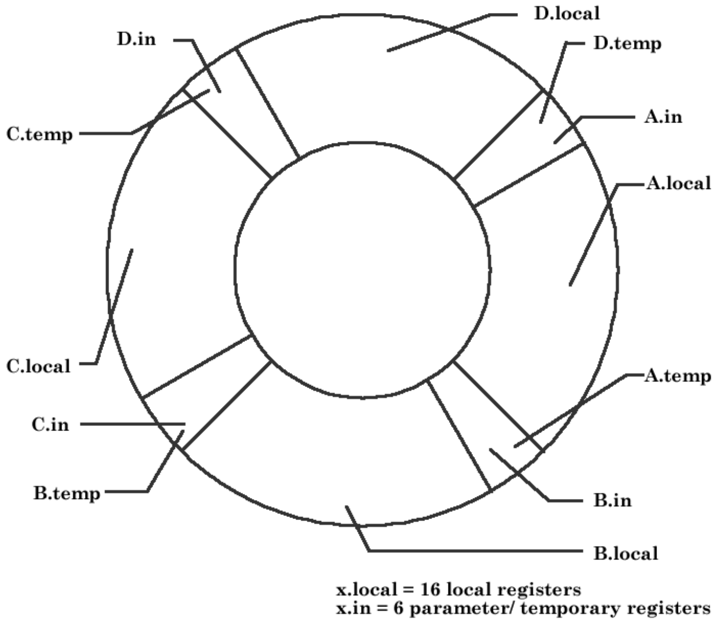
Fall 2006 CSCI 4717 – Computer Architecture Due: Tuesday, December 5, 2006

RISC Homework



1.) Assume that a RISC processor uses register windows configured like that shown in the figure to the right. To what depth can a process call procedures before having to push a window of registers to the stack, i.e., how many function calls can be present at one time? (Assume that the process in window A is at a depth of 1.)

2.) For the figure in problem 1, if a process is called that requires pushing a window to the stack, how many registers must be pushed?

3.) Below is the timing diagram for the execution of 5 instructions on a RISC processor with 2-way pipelined timing, i.e., I and E stages are executed simultaneously, but bus limitations require D to be executed alone. How many cycles would be saved if a dual-port RAM is used permitting two memory accesses at one time?

LOAD A←M	**I**	**E**	**D**								
ADD C←A+5		**I**		**E**							
LOAD B←M				**I**	**E**	**D**					
COMPARE B←C					**I**		**E**				
BRANCH EQ X							**I**	**E**			
NOOP								**I**	**E**		
STORE M←C									**I**	**E**	**D**

4.) Find the absolute minimum number of registers required to execute the code below. (Hint: Think like an assembly language programmer trying to minimize the number of registers needed for the code.)

```
int done = 0;
int user_input << cin;
while (!done)
{
   int calc = 0;
   switch (user_input)
   {
      case 0:
         for (int i=0; i<5; i++) calc = (calc << 1) + i;
         break;
      case 1:
         for (int j=0; j<10; j++) calc = (calc << 1) + j;
         break;
      default:
         calc = 7;
         break;
   }
   for (int k=0; k<calc; k++) cout << ".";
}
```