|  |
| --- |
| **Assignment # 6**  **SYSC 5704 – Elements of Computer Systems** |
| Fall 2014  Submitted To  Dr. R. Gregory Franks  By  **Ferhan Jamal (100 953 487)**  Carleton University |

**1. [5] <§6.2> Answer**:-

In the 1st part of the step 3, it is given that we have that assume that we have one oven large enough to hold one cake, one large bowl, one cake pan, and one mixer, and we have to come up with a schedule to make three cakes as quickly as possible and also we have to identify the bottlenecks in completing this task.

The only thing that can occur in parallel is the oven heating until the cake pan is loaded and baking starts giving an equal appropriate time value to each (as a symbol) but the single bowl, single mixer, single cake pan, and single oven are the bottlenecks for the tasks associated with them. The single user can also be a bottleneck for the mixing and also to speed up the tasks.

**2. [5] <§6.2> Answer:-**

In the 2nd part of the step 3, we have 3 bowls, 3 cake pans, and 3 mixers but still we have one user to do all the tasks.

The only thing that can be possible is that to leave the mixers running at the appropriate speeds in parallel, so the mixing can happen in parallel.

**1. [10] <§6.2>Answer:-**

A single iteration of the loop will take:

20 cycles (6 + 6 + 4 + 1 + 2 + 1)

**2. [10] <§6.2>Answer:-**

The loop carried dependencies in the above code are on :

D[j-1] and D[j-2] ( These 2 are in the iteration which calculates D[j] )

The assembly level registers would be $f0 and $f2 ( These 2 are in the add.d instruction that calculates $f4 )

**3. [10] <§6.2> Answer:**

The code will execute as follows, after unrolling it 4 times in the code:-

for( j = 2; j < 996; j = j + 4)

{

D[j] = D[j-2] + D[j-1];

D[j+1] = D[j-1] + D[j];

D[j+2] = D[j] + D[j+1];

D[j+3] = D[j+1] + D[j+2];

}

Let us assume the fact that each of the processors have their own registers (i.e.2 of the processor have their own every $s1) then the code will then look like as below:

At node :

addiu $s2,$zero,7968 #8000-32

addiu $s1,$zero,16

l.d $f0,-16($s1) #first time load of D[j-2]

l.d $f2,-8($s1) #first time load of D[j-1]

loop: add.d $f4,$f0,$f2 #first addition D[j] = D[j-2] + D[j-1]

add.d $f5,$f2,$f4 #second addition D[j+1] = D[j-1] + D[j]

send (2, $f4) #send D[j] to node 2

send (2, $f5) #send D[j+1] to node 2

s.d $f4,0($s1) #store D[j]

s.d $f5,8($s1) #store D[j+1]

receive ($f0) #receive D[j+2] into $f0, therefore no loading is necessary

receive ($f2) #receive D[j+3] into $f2, therefore no loading is necessary

s.d $f0,16($s1) #store D[j+2]

s.d $f2,24($s1) #store D[j+3]

addiu $s1,$s1,32 #stride is 32 (4 elements)

bne $s1,$s2,loop

On node 2,

addiu $s1,$zero,0

addiu $s2,$zero,2000 #8000/4 This loop will happen 2000 times

loop: receive ($f1) #receive D[j]

receive ($f2) #receive D[j+1]

add.d $f3,$f1,$f2 #third addition D[j+2] = D[j] + D[j+1]

add.d $f4,$f2,$f3 #fourth addition D[j+3] = D[j+1] + D[j+2]

send (1,$f3) #send D[j+2] to node 1

send (1,$f4) #send D[j+2] to node 1

addiu $s1,$s1,1

bne $s1,$s2,loop

**4. [10] <§6.2>Answer:-**

To obtain any speedup from the message passing distributed system described above the interconnected network should have to respond or execute in a onesingle cycle which is practically impossible

**1. [10] <§6.11> Answer:-**

For Row 1:

In 1ms, there is one transaction so the no. of requests per second will be 1000

The maximum transaction processing rate is 5000/sec so the maximum number of transactions which the quad-core system can possess is 5000/sec.

For Row 2:

In 2ms, there is one transaction so the number of requests per second will be 500

The maximum transaction processing rate is 5000/sec so the maximum number of transactions which the quad-core system can possess is 5000/sec.

For Row 3:

In 1ms, there is one transaction so the no. of requests per second will be 1000

The maximum transaction processing rate is 10,000/sec so the maximum number of transactions which the quad-core system can possess is 10,000/sec.

For Row 4:

In 2ms, there is one transaction so the number of requests per second will be 500

The maximum transaction processing rate is 10,000/sec so the maximum number of transactions which the quad-core system can possess is 10,000/sec.

**2. [10] <§6.11>Answer:-**

If there is an 8-core system, then the throughput will be double generally.

**3. [10] <§6.11>Answer:-**

We rarely obtain this kind of speed-up by simply increasing the number of cores because of the chances of paging in the memory and also there may be some issues related to scheduling(order in assigned jobs execute) and synchronization which in-turn effects the workload and the speed up. We calculate speed-up just to know how much performance we are getting by increasing the number of processors.