CMPT 300 - Assignment 2

Jin Young Kim (301201074)
Injun Son (301374509)
Insoo Rhee (301075548)

Problem 1.

a) [10 points] What is the maximum value of tally? Give an example of an order of execution of Half() and Double() that would produce the maximum value.

Assume Thread 1 = double() and Thread 2 = half(). Then, the maximum value of tally will be 4096 x 2^12.

1. Thread 1 runs until count = 10. Therefore, it loads **tally = 4096 x 2^11** into memory.
2. Thread 1 runs for count = 11. Therefore, **tally = 4096 x 2^12**. However, Thread 2 interrupts Thread 1 before this value can be stored in memory.
3. Thread 2 runs to completion. Therefore, tally = **4096 x 2^11 / 2^12 = 2048** in memory.
4. Thread 1 resumes, and stores the value of 4096 x 2^12 to memory.
5. Therefore, the global variable tally is now **4096 x 2^12**

b) [10 points] What is the minimum value of tally? Give an example of an order of execution of Half() and Double() that would produce the minimum value.

With the same assumptions as above, the minimum value of tally will be 1.

1. Thread 2 runs until count = 10. Therefore, it loads **tally = 4096 / 2^11** into memory.

2. Thead 2 runs for count = 11. Therefore, **tally = 4096 / 2^12 = 1**. However, Thread 1 interrupts Thread 2 before this value can be stored in memory.
3. Thread 1 runs to completion. Therefore, **tally = 4096 / 2^11 * 2^12 = 8192** in memory.
4. Thread 2 resumes, and stores the value of 1 to memory.
5. Therefore, the global variable tally is now **1.**

c) [15 points] What order could the processes run in to produce a value of the global variable tally of 1024 after both processes had completed?

We start with tally = 4096, also equivalent to 2^12. We want our tally to be 2^10 after the two threads run to completion.

1. Thread 1 enters the for loop and doubles tally until count is 9. It writes the value of tally back to the variable tally. Tally is now 2^22.
2. The scheduler replaces Thread 1 with Thread 2. Thread 2 enters the for loop and halves tally until count is 11, but doesn't get to write the final value of tally to the variable tally. Therefore, the value of tally inside Thread 1's register is now 2^10, and that inside the memory is 2^11.
3. The scheduler replaces Thread 2 with Thread 1, which runs to completion and writes the value of 2^13 to the variable tally.
4. Thread 1 resumes, writing the value of 2^10 to the variable tally.
5. The variable tally in memory now has a value of 2^10.

Thread1				Thread2			
Value copied into register	Value in register after multiplication	value copied back into variable	count	Value copied into register	Value in register after division	value copied back into variable	count

2^12	2^13	2^13	0				
2^13	2^14	2^14	1				
2^14	2^15	2^15	2				
2^15	2^16	2^16	3				
2^16	2^17	2^17	4				
2^17	2^18	2^18	5				
2^18	2^19	2^19	6				
2^19	2^20	2^20	7				
2^20	2^21	2^21	8				
2^21	2^22	2^22	9				
Swap to Thread 2							
				2^22	2^21	2^21	0
				2^21	2^20	2^20	1
				2^20	2^19	2^19	2
				2^19	2^18	2^18	3
				2^18	2^17	2^17	4
				2^17	2^16	2^16	5
				2^16	2^15	2^15	6
				2^15	2^14	2&14	7
				2^14	2^13	2^13	8
				2^13	2^12	2^12	9
				2^12	2^11	2^11	10

				2^11	2^10		11
				Swap to thread 1			
2^11	2^12	2^12	10				
2^12	2^13	2^13	11				
Swap to thread 2							
						2^10	11