CSE 321a: Computer Organization (I)
Third Year, Computer & Systems Engineering

Solution to Assignment #1

1. The performance of a single-processor machine is evaluated using a 2-program benchmark suite …

Suppose for all values of N (where N is positive integer):
Fraction of first program that can be executed in parallel = $f_A = 0.5$
Fraction of second program that can be executed in parallel = $f_B = 0.875$
Time to execute the first program by an N-processor machine = T_{AN}
Time to execute the second program by an N-processor machine = T_{BN}
Time to execute the first program by the reference machine = T_{AR}
Time to execute the second program by the reference machine = T_{BR}
Benchmark score of an N-processor machine = B_N

$$B_N = \sqrt{\left(\frac{T_{AR}}{T_{AN}}\right)\left(\frac{T_{BR}}{T_{BN}}\right)}$$

Factor of improvement in benchmark score for N-proc. machine (over 1-proc. machine) = F_N

$$\boldsymbol{F_N} = \frac{\boldsymbol{B_N}}{\boldsymbol{B_1}} = \frac{\sqrt{\left(\frac{T_{AR}}{T_{AN}}\right)\left(\frac{T_{BR}}{T_{BN}}\right)}}{\sqrt{\left(\frac{T_{AR}}{T_{A1}}\right)\left(\frac{T_{BR}}{T_{B1}}\right)}} = \sqrt{\left(\frac{T_{A1}}{T_{AN}}\right)\left(\frac{T_{B1}}{T_{BN}}\right)} = \sqrt{\left(\frac{1}{1-f_A+\frac{f_A}{N}}\right)\left(\frac{1}{1-f_B+\frac{f_B}{N}}\right)}$$

$$= \sqrt{\left(\frac{1}{0.5+\frac{0.5}{N}}\right)\left(\frac{1}{0.125+\frac{0.875}{N}}\right)}$$

(a) What is the maximum factor of improvement that can be achieved in the benchmark score …?

Maximum factor of improvement can be achieved when $N \rightarrow \infty = F_\infty$

$$\boldsymbol{F_N} = \sqrt{\left(\frac{1}{0.5+\frac{0.5}{\infty}}\right)\left(\frac{1}{0.125+\frac{0.875}{\infty}}\right)} = \sqrt{(2)(8)} = 4$$

(b) What is the minimum number of processors that need to be added to that machine ...?

We need to find the value of N that makes $F_N = 3$

$$\boldsymbol{F_N} = \sqrt{\left(\frac{1}{0.5+\frac{0.5}{N}}\right)\left(\frac{1}{0.125+\frac{0.875}{N}}\right)} = \sqrt{\left(\frac{2N}{N+1}\right)\left(\frac{8N}{N+7}\right)} = 3$$

$$\frac{16N^2}{(N+1)(N+7)} = 9$$

$$7N^2 - 72N - 63 = 0$$

Roots of this quadratic equation are 11.1 and -0.8
This means that the minimum value of N required to triple the benchmark score is 12
In other words, we need to **add 11 more processors** to that machine in to triple its benchmark score!!

2. **In a hypothetical computer, the processor has five registers: a 20-bit Program Counter (PC), …**
   (a) In this hypothetical computer:

   i. Which memory locations can be used to store instructions?

   PC is 20-bit long ➔ Instructions can be stored in memory locations 00000 to FFFFF.

   ii. Which memory locations can be used to store data?

   Address field in instructions is 13-bit long ➔ Data can be stored in memory locations 0000 to 1FFF.

   iii. What is the maximum number of loop iterations (without nesting)?

   CTR is 10-bit long ➔ maximum number of loop iterations = 2^{10} = 1024

   (b) Show, using the table below, the execution trace of that program …

Instruction	Cycle	PC	AC	CTR	PTR1	PTR2	Location: 005D7	Location: 00B9F	Location: 00BA0	Location: 01CE3	Location: 01CE4
Initially	---	B739C	3C0DE	2F1	1FFF	100D	00000	20005	20020	00007	2001B
13C02	Fetch	B739D	3C0DE	2F1	1FFF	100D	00000	20005	20020	00007	2001B
	Execute	B739D	3C0DE	002	1FFF	100D	00000	20005	20020	00007	2001B
03CE3	Fetch	B739E	3C0DE	002	1FFF	100D	00000	20005	20020	00007	2001B
	Execute	B739E	3C0DE	002	1CE3	100D	00000	20005	20020	00007	2001B
04B9F	Fetch	B739F	3C0DE	002	1CE3	100D	00000	20005	20020	00007	2001B
	Execute	B739F	3C0DE	002	1CE3	0B9F	00000	20005	20020	00007	2001B
0DFFF^{+}	Fetch	B73A0	3C0DE	002	1CE3	0B9F	00000	20005	20020	00007	2001B
	Execute	B73A0	2000C	002	1CE3	0B9F	00000	20005	20020	00007	2001B
345D7^{+}	Fetch	B73A1	2000C	002	1CE3	0B9F	00000	20005	20020	00007	2001B
	Execute	B73A1	2000C	002	1CE3	0B9F	00090	20005	20020	00007	2001B
09FFF	Fetch	B73A2	2000C	002	1CE3	0B9F	00090	20005	20020	00007	2001B
	Execute	B73A2	2000C	002	1CE4	0BA0	00090	20005	20020	00007	2001B
16004	Fetch	B73A3	2000C	002	1CE4	0BA0	00090	20005	20020	00007	2001B
	Execute	B739F	2000C	001	1CE4	0BA0	00090	20005	20020	00007	2001B
0DFFF^{+}	Fetch	B73A0	2000C	001	1CE4	0BA0	00090	20005	20020	00007	2001B
	Execute	B73A0	20005	001	1CE4	0BA0	00090	20005	20020	00007	2001B
345D7^{+}	Fetch	B73A1	20005	001	1CE4	0BA0	00090	20005	20020	00007	2001B
	Execute	B73A1	20005	001	1CE4	0BA0	000A9	20005	20020	00007	2001B
09FFF	Fetch	B73A2	20005	001	1CE4	0BA0	000A9	20005	20020	00007	2001B
	Execute	B73A2	20005	001	1CE5	0BA1	000A9	20005	20020	00007	2001B
16004	Fetch	B73A3	20005	001	1CE5	0BA1	000A9	20005	20020	00007	2001B
	Execute	B73A3	20005	000	1CE5	0BA1	000A9	20005	20020	00007	2001B
285D7*	Fetch	B73A4	20005	000	1CE5	0BA1	000A9	20005	20020	00007	2001B
	Execute	B73A4	000A9	000	1CE5	0BA1	000A9	20005	20020	00007	2001B

365D7*	Fetch	B73A5	000A9	000	1CE5	0BA1	000A9	20005	20020	00007	2001B
	Execute	B73A5	000A9	000	1CE5	0BA1	0000D	20005	20020	00007	2001B
3FFFF	Fetch	B73A6	000A9	000	1CE5	0BA1	0000D	20005	20020	00007	2001B
	Execute	B73A6	000A9	000	1CE5	0BA1	0000D	20005	20020	00007	2001B

(c) What does the program compute?

This program calculates the **distance between two points** (whose (*x*,*y*) coordinates are stored in memory starting at locations 00B9F and 01CE3) and saves the result to location 005D7.

(d) How much time does it take for the program to be executed? Justify your answer.

This hypothetical machine has three types of instructions:

1. Instructions that don't have memory operands (e.g., 09FFF and 16004). Each of these instructions takes 5 cycles: 3 (instruction fetch) + 2 (instruction processing).
2. Instructions that have one memory operand (e.g., 285D7 and 365D7). Each of these instructions (marked by '*' in the table) takes 8 cycles: 3 (instruction fetch) + 2 (instruction processing) + 3 (operand fetch or result store).
3. Instructions that have two memory operands (e.g., 0DFFF and 345D7). Each of these instructions (marked by '+' in the table) takes 11 cycles: 3 (instruction fetch) + 2 (instruction processing) + 3 (operand fetch) + 3 (operand fetch or result store).

Total number of cycles
= (8 instructions) * (5 cycles) + (2 instructions) * (8 cycles) + (4 instructions) * (11 cycles)
= 100 cycles

Clock period = 1 / (50 MHz) = 20 nS

Total time = (100 cycles) * (20 nS) = 2 μS