Skill Assessment #6

● Graded

Student

HARRY KIM

Total Points

96 / 100 pts

Question 1

Modified CPU - lui instruction

40 / 40 pts

✔ **– 0 pts** 1 Correct.

– 2 pts Added an extra mux on part 1

– 4 pts 1 Slightly Incorrect

– 8 pts 1 Fully Incorrect

✔ **– 0 pts** 2 Correct

– 8 pts 2 Missing

✔ **– 0 pts** 3 Correct

– 4 pts 3 Slightly Incorrect

– 8 pts 3 Fully Incorrect

✔ **– 0 pts** 4 Correct

– 2 pts Minor mistake in part 4

– 4 pts 4 Slightly Incorrect

– 8 pts 4 Fully Incorrect

✔ **– 0 pts** 5 Correct

– 2 pts 5 Slightly Incorrect

– 6 pts 5 Fully Incorrect

– 4 pts Handwritten Diagram

Question 2

Modified CPU - jsa instruction

36 / 40 pts

- ✔ – 0 pts 1 Correct
- – 4 pts 1 Slightly Incorrect
- – 8 pts 1 Fully Incorrect
- ✔ – 0 pts 2 Correct
- – 8 pts 2 Fully Incorrect
- ✔ – 0 pts 3 Correct
- – 4 pts 3 Slightly Incorrect
- – 8 pts 3 Fully Incorrect
- ✔ – 0 pts 4 Correct
- – 4 pts 4 Slightly Incorrect
- – 8 pts 4 Fully Incorrect
- – 0 pts 5 Correct
- ✔ – 4 pts 5 Slightly Incorrect
- – 8 pts 5 Fully Incorrect
- – 4 pts Handwritten Diagram
- 💬 JSA 1, Dst X, Mem read 1 Mem Write 0

Question 3

Original CPU - Clock period

20 / 20 pts

- – 5 pts Wrong Instruction
- – 3 pts Very Wrong on Picoseconds
- – 5 pts Incorrect Diagram
- – 5 pts Didn't calculate Freq
- ✔ – 0 pts Correct
- – 2 pts Very wrong on Freq
- – 20 pts Not attempted

Question 4

Late Penalty (if appropriate)

0 / 0 pts

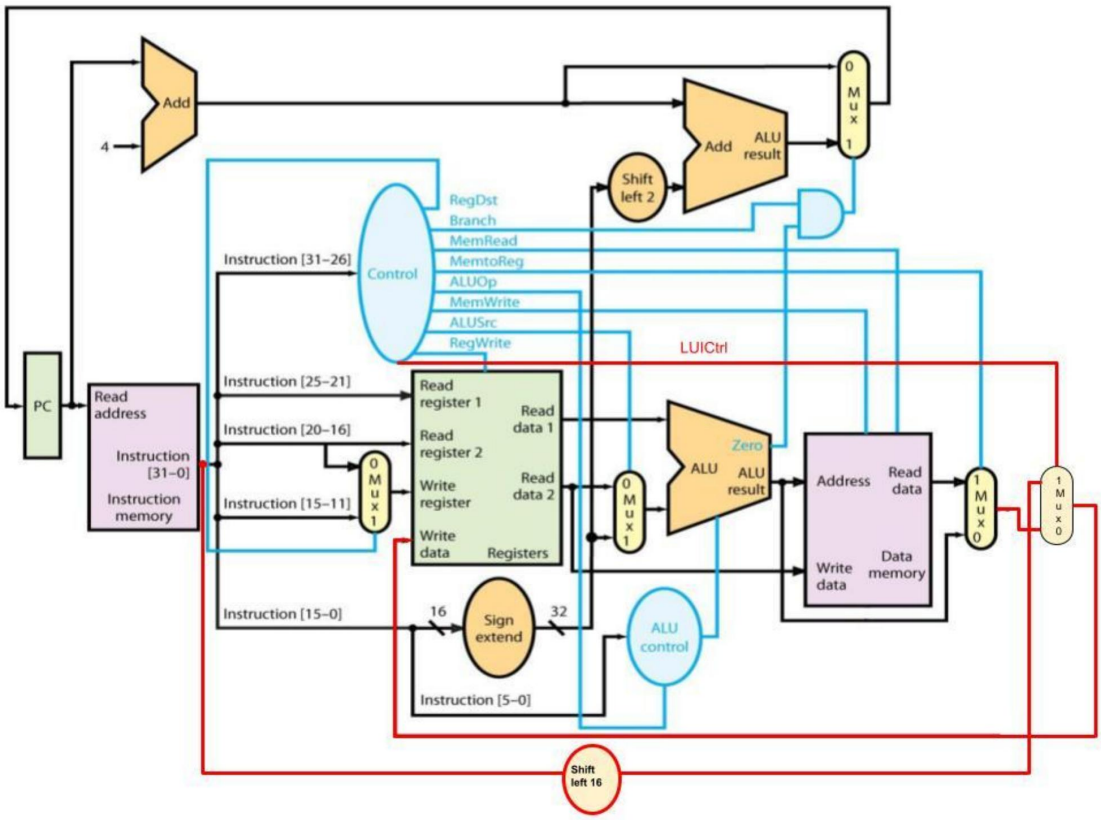
- ✔ + 0 pts Correct
- – 10 pts Click here to replace this description.

Question assigned to the following page: 1

Harry Kim
u1226472

Skill Assessment #6

A1.1:



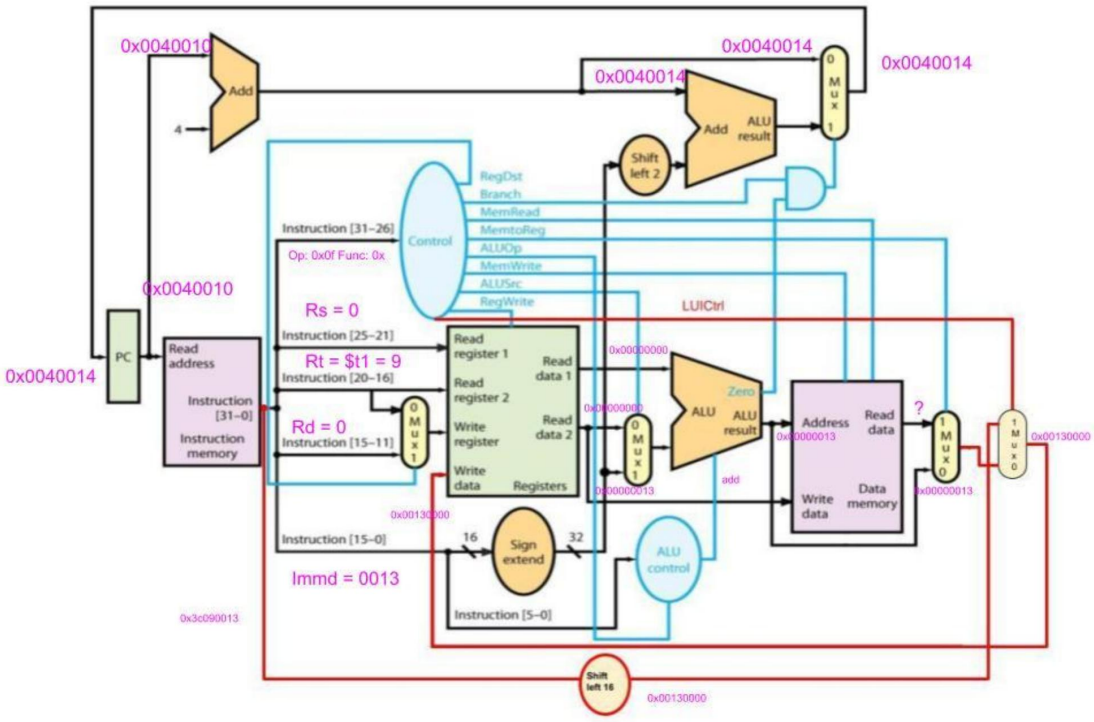
A1.2: Load upper immediate (lui) loads the first 16 bits of a register while the rest of the 16 bits are set to zero. By taking the instruction bits and shifting left by 16 bits, we are left with a 32 bit instruction with the 16 most significant bits set with data from the original instruction with the 15 least significant bits set to zero. We then can send this data to a multiplexor allowing us to choose which data we send back to the registers to write the data.

Question assigned to the following page: 1

Harry Kim
u1226472

A1.3:

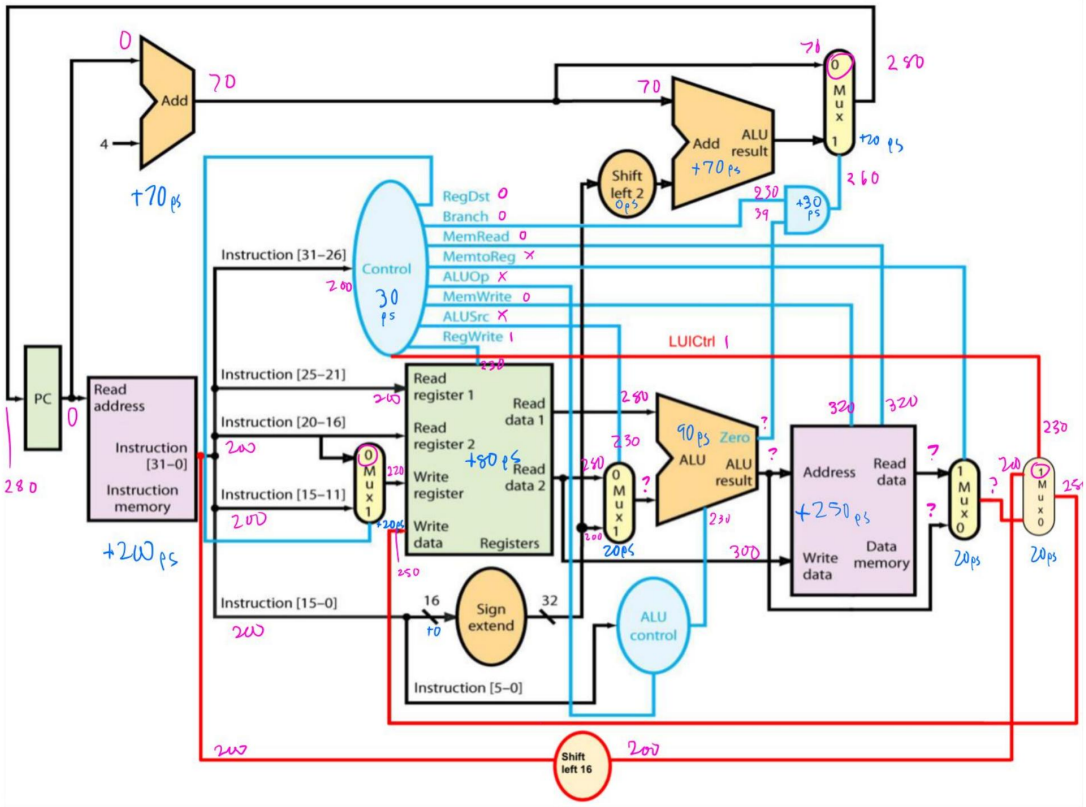
lui $t1, 0x0013



Question assigned to the following page: 1

Harry Kim
u1226472

A1.4:



After writing out all the delays the CPU would have (using the assumptions from Part 3 of the study question) computing an lui instruction, we can see that the longest delay is actually in inputting the next instruction address into the program counter. Getting the upper immediate and writing the data to the register takes less time than getting the next instruction address. Since it arrives at +280 ps, and since we need to allow a 10 ps setup time, the clock cycle must be *at least* 290 ps.

Question assigned to the following page: 1

Harry Kim
u1226472

A1.5:

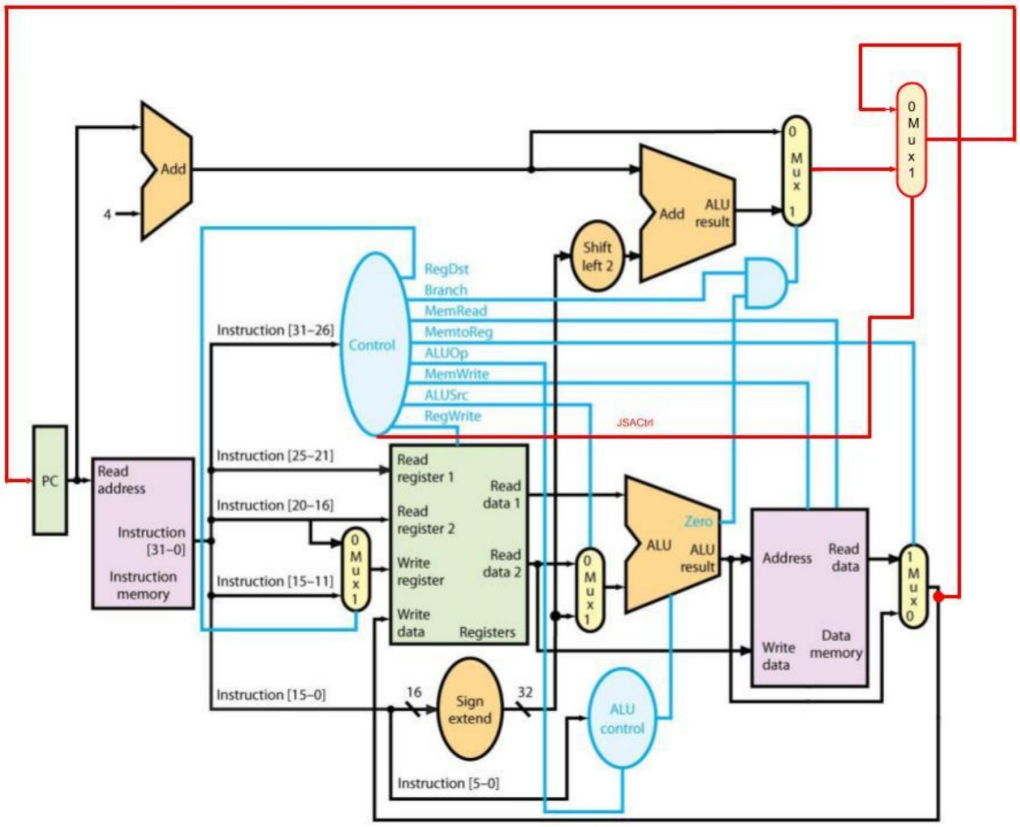
Control lines

ALU src	x
Reg Dst	0
Branch	0
MemRead	x
MemWrite	0
RegWrite	1
LUICtrl	1
MemtoReg	x
ALUop	x

Question assigned to the following page: 2

Harry Kim
u1226472

A2.1:



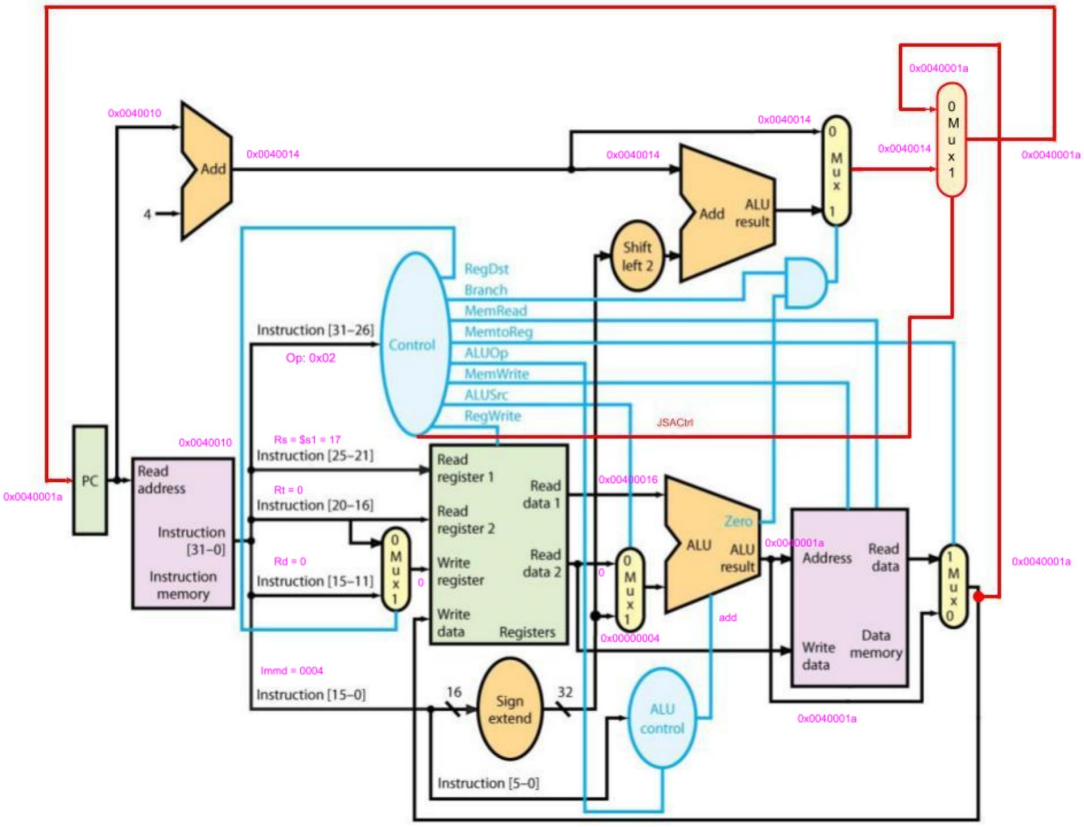
A2.2: The jsa instruction jumps to an address that is stored in memory. Being that the jsa is an I-type instruction, and jsa being similar to a lw and sw instruction, the address must first be read from a register and then added by its offset. Instead of storing the information back into the registers, we can take a wire and use the data, which is the new address, input it into a multiplexer that chooses either the new address or the next sequential address, then input it directly into the program counter.

Question assigned to the following page: 2

Harry Kim
u1226472

A2.3:

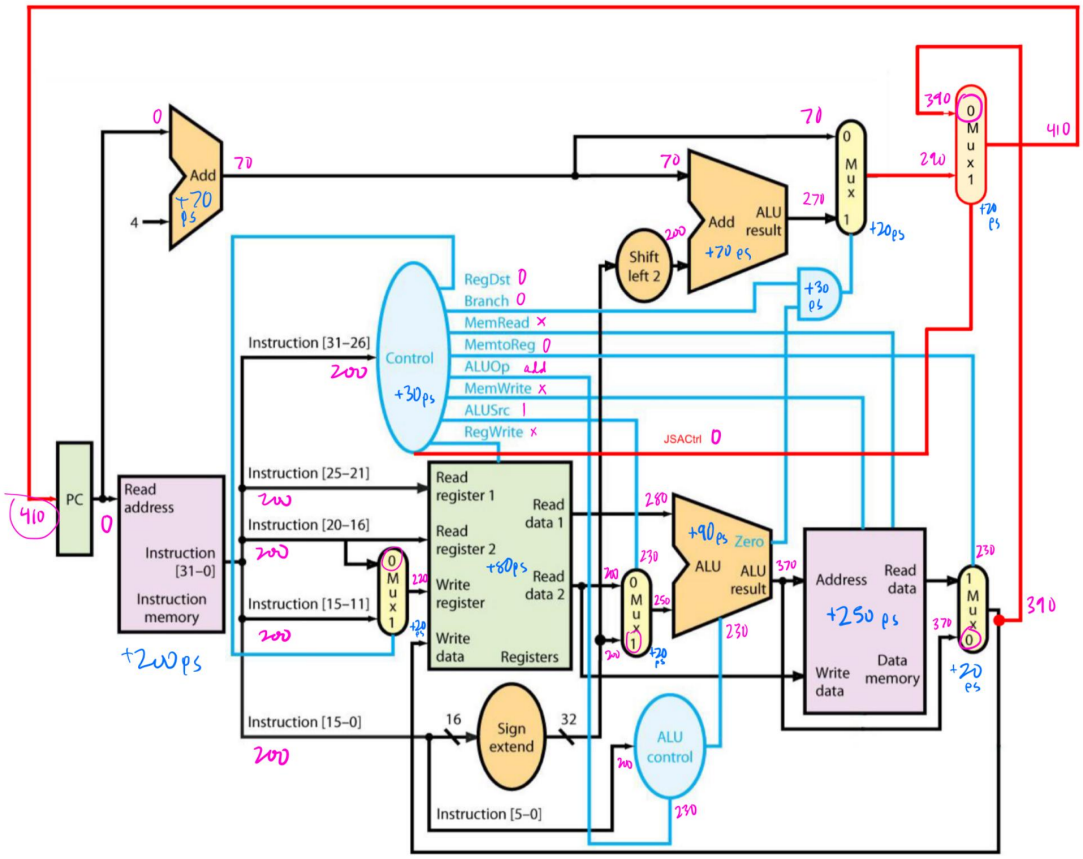
$s1 holds 0x00400016
jsa 4($s1)



Question assigned to the following page: 2

Harry Kim
u1226472

A2.4:



After writing out all the delays the CPU would have (using the assumptions from Part 3 of the study question) computing an lui instruction, we can see that the longest delay is inputting the next instruction address into the program counter. This makes sense as the address to jump to needs to be read from a register and added by the offset. Since it arrives at +410 ps, and since we need to allow a 10 ps setup time, the clock cycle must be *at least* 420 ps.

Question assigned to the following page: 2

Harry Kim
u1226472

A2.5:

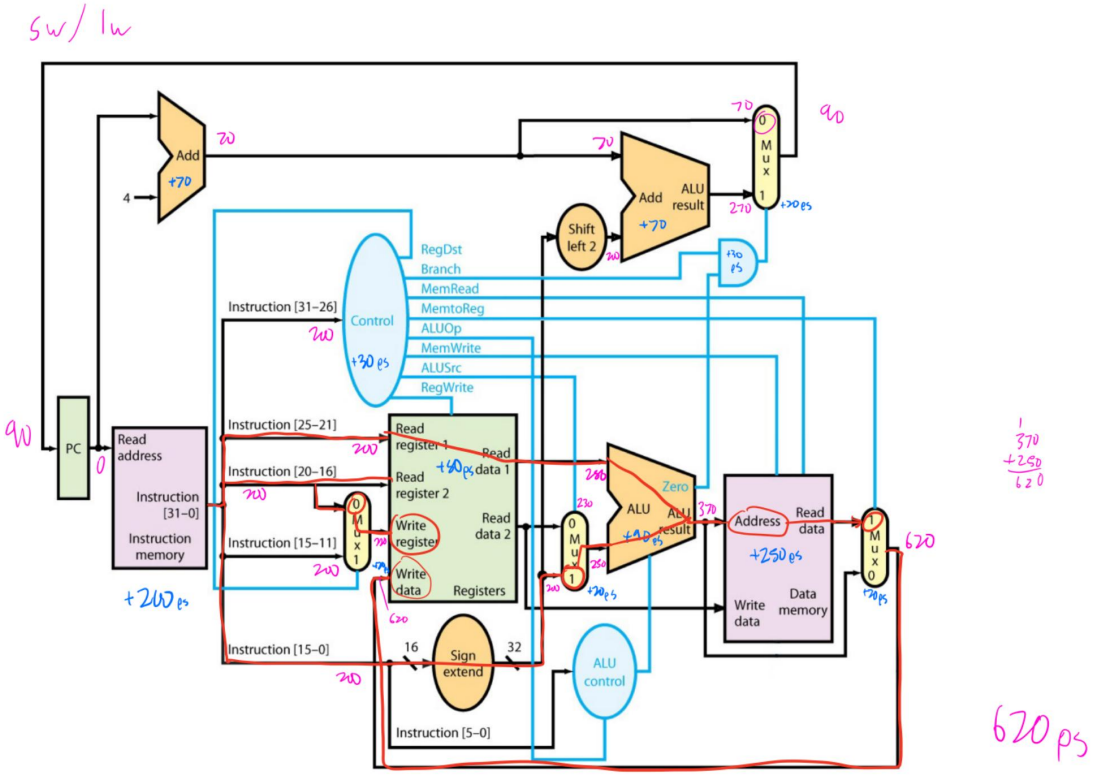
Control lines

ALU src	1
Reg Dst	0
Branch	0
MemRead	x
MemWrite	x
RegWrite	x
JSACtrl	0
MemtoReg	0
ALUop	add

Question assigned to the following page: 3

Harry Kim
u1226472

A3: When executing an arbitrary mix of instructions, every wire, logic gate, and logic block eventually computes and outputs a value. Since we don't know what the next instruction will be, we need to assume that the instruction with the longest clock period and clock frequency is the fastest at which our single cycle CPU could safely operate at. The list of instructions that this CPU currently supports is add, addi, sub, and, or, nand, nor, slt, lw, sw, and beq. After writing the latency of each instruction, lw and sw take the longest because they must read the data from the address they provide which takes the most significant amount of time on top of reading from registers and adding the data from the register and the immediate to get the address. This process takes 620 ps (using the assumptions from Part 3 of the study question).



With a 10% variance in the clock period, the worst case delay would be 620 * 110% or 682 ps. The clock frequency would be 1.466 GHz.