|  |  |  |
| --- | --- | --- |
| 1a)  b) | Consider the following 4 bit multiplication unit. Briefly explain how it works. (You can consider drawing flowchart to explain it) Idenitfy the drawbacks of this method.    There is a faster method of multiplication known as loop unrolling. Draw the loop unrolled version for the above multiplication unit. | 5  5 |
| 2. | Translate the following two separate C Codes into MIPS code   1. y = h + g + 1   (y, g and h are in $S0, $S1 and $S2 )   1. A[3] = 2 x A [i]   (Base address of A is in $S0, i is in $S1) | 3+3 |
| 3. | Explain with an example :  a) How NOR instruction can be used for complementing a data  b) How ADD instruction can be used for moving data | 2+2 |
| 4. | Compare stack and accumulator based architecture. | 5 |
| 5. | Suppose we developed a new, simpler processor that has 80% of the capacitive load of the more complex older processor. Further, assume that it has adjustable voltage so that it can reduce voltage 5% compared to processor B, which results in a 20% shrink in frequency. What is the impact on dynamic power? | 5 |
| 7. | You have been asked to design a small computing system with a customized 8 bit ISA that can execute the following types of program. You can assume that all the variables are already loaded in some registers.  ***You should consider using only the instructions covered in the lecture***  if ( i < Sum){  Sum = Sum \* 64  i = i \* 64  }  else  i = i \* 2  **Answer the following designing questions for the most optimized solution**  1) How many formats would you choose? Why?  2) Draw the formats along with field name and number of bits in each field.  3) Draw a register table. (with register name, values and behavior)  4) Draw a table with list of instructions, instruction type, their opcode, functionality (if any) etc.  5) Translate the above HLL code into assembly code based on your ISA | 15 |
|  | (Rough) |  |