IIT JODHPUR

Minor Examination: Processor Design (Sept'24) [OPEN-BOOK]

Guidelines (Total time: 120 minutes, Maximum Marks: 25):

- Please read the question paper very carefully. **NO clarification is required in any question. Under NO Circumstances, You are permitted to ask any doubt to the invigilator.** In case of any doubt, assume whatever you wish to and state that in your answer.

---

1. Most edge computing applications require specialized/custom hardware designs. However, domains such as IoT may involve different types of applications. One way this could be done is through design of a tiny processor (supporting only a few instructions and having minimal hardware). Consider that the below figure denotes the twelve instructions supported (on the LHS) and a sample C program (on the RHS). Design the **datapath for a non-pipeline implementation** given the following constraints? Note that the program/application to be run on this processor could be any program (similar to the sample program in below figure). [5 + 2 Marks]

   - You are allowed to use a simple ALU that performs **only ADD and SUB instructions**.
   - The register file has 11 32-bit registers, out of which R0 to R7 are 8 general-purpose registers whereas remaining can be for stack pointer, link register and program counter.
   - The flag register is a special register having only 2 bits for 2 flags (Carry and Zero).
   - Consider an instruction memory and program memory as per your choice.
   - You can use a few logic blocks like multiplexers/comparators to complete your datapath.

SUPPORTED INSTRUCTIONS ON PROCESSOR	Sample Program

ADD Rk, Rj, Ri
SUB Rk, Rj, Ri
MUL Rk, Rj, Ri
DIV Rk, Rj, Ri
ADDC Rk, Rj, Ri //ADD if carry flag is set
OR Rk, Rj, Ri // Rk = Rj || Ri
INC Rk // Increment
DEC Rk //Decrement
LD Rk, Rj // Rk = Mem[Rj]
ST Rj, Rk // Mem[Rj] = Rk
BZ Rk // Branch if zero to address given in Rk
BC Rk // Branch if carry to address given in Rk
	int main() {
	
	int a, b, c, d, e, f, g, i;
	for (i=100; i>0; i- -){
	
	c = a + b;
	d = d + c + 5a;
	e = d/b;
	f = a x e;
	if (f > d)
	g = a || b ;
	return 0; }

   You can explain the execution of any 2 instructions (not of same type) on the datapath designed by you. Example- You may explain the execution of ADD and BZ instructions.

2. Consider the design that you have attempted in Question 1. You are asked to implement a pipelined version of the design to achieve high performance. So, you need to modify the design in Question 1 to accommodate all the possible pipeline hazards that may arise and achieve higher performance than the original design. Design the **control path (giving all the required control words) for this pipelined implementation**. You can consider a typical 5-stage pipeline (Fetch, Decode/RegRead, Execute, MemAccess, WriteBack). Please measure the average performance speed-up that you can obtain in comparison to Question 1? [4 + 2 Marks]

3. Consider that you are asked to design an on-chip branch predictor (BP) to be integrated with the pipelined processor implementation in Question 2 to improve its performance. You decide to implement a 2-bit branch predictor (that has 4 states) for this purpose. Measure quantitatively the speed-up in the performance that you can obtain in this redesigned pipeline processor over the design you have completed in Question 2 above (which is without BP) ? [3 Marks]

4. To achieve further higher performance in comparison to pipeline processors for vector type workloads, a Vector Unit (VU) needs to be designed for the implementation carried out in Question 1 and Question 2. Design a VU that can be integrated with the pipeline processor implementation you have carried out in Question 2 above. Also, measure the speed-up in the performance with the VU-integrated processor design as compared to the pipelined processor implementation in Question 2 for the loop (part of program) shown below. [4 + 3 Marks]

```
for( j=0; j <=99; j++){
C[j] = A[j] + B[j];
D[j] = C[j] + 6A[j];
E[j] = D[j]/B[j];
F[j] = A[j]xE[j]; }
```

   You can note the following considerations for your VU design and measuring its performance:

   - Recall that for your processor design, you have implemented a datapath in Question 1. Would you like to modify that datapath design for the VU? Note that the constraints mentioned in Question 1 apply to this Question 4 also. So, you do not have vector multiplication/division ALUs, you have access to only that simple ALU replaced with vector counterpart.
   - You can consider that you have vector registers (of vector length as 100) available to you (vector counterpart of registers mentioned in Question 1).
   - As you are responsible for the design of datapath (in Question 1 and Question 4), please calculate the latency of your datapath for all arithmetic-type instructions.
   - You can consider 8 parallel access to a banked memory (word-interleaved) and the memory read/write latency is equal to 10 cycles.

5. Because of certain changes in the compiler, the number of instructions gets doubled for a program. Additionally, the clock frequency gets halved because of changes in manufacturing technology node. The execution time of this program in a computer architecture simulator (modeling the above conditions) gets changed by a factor of S. What is S? [2 Marks]
