EECS 151/251A Homework 7

Due Monday, April 3, 2023

Problem 1: Tradeoffs

The execution time of program is expressed as

$$\#instructions \times CPI \times Clock\ period. \quad (1)$$

Imagine we have designed a microarchitecture of 3-stage RISC-V CPU, which resolves all hazards by stalling. For each of the following changes, explain its effect over #instructions, CPI, clock period, and overall performance.

1. Use CISC (e.g. memory-to-memory instructions) instead of RISC. Assume we maintain the clock period.
2. Implement forwarding.
3. Implement a branch predictor.
4. Implement 5-stage pipelining.
5. Implement superscaler (execute multiple instructions at a time).
6. Implement cache (assume the original microarchitecture does memory access in one cycle).

Solution:

#instructions is not affected by other than the first change.

1. #instructions decreases. CPI increases (memory-to-memory instructions require stalls to maintain the clock period). Performance is likely to degrade as it does not exploit locality (in RISC, register file works as a temporary storage, which reduces memory access).
2. CPI decreases for less stalling for data hazards. The clock period may increase. Performance would improve unless the forwarding path becomes the critical path.
3. CPI decreases for less stalling for control hazards. The clock period is likely to stay the same (ALU or memory paths should have a larger delay). Perforamnce is likely to improve.
4. CPI increases a little due to more hazards. The clock period decreases. Performance is likely to improve. (We will give full credits to answers with different overall performance estimate —degrade or undeterministic— if some reasoning is provided. Example: the clock frequency will increase by at most 5/3, while we need one more bubbles for existing hazards and a new bubble for 3-cycle apart data hazards. Let p be the probability of control or 1-cycle apart data hazards, q for 2-cycle apart data hazards, and r for 3-cycle

apart data hazards. Then CPI will increase from $1 + 2p + q$ to $1 + 3p + 2q + r$. Assuming q and r are small (bubbles injected for other hazards may resolve them as well), the relative CPI $(1 + 3q)/(1 + 2q)$ can take a value between 1 and 4/3. Depending on critical paths and hazard probabilities, this could overwhelm the frequency increase.)

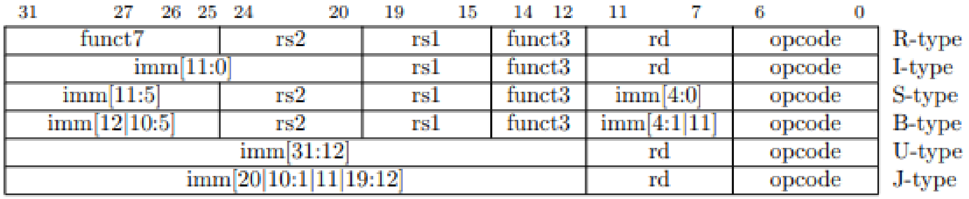
5. CPI decreases. The clock period increases. Performance is likely to improve.
6. CPI increases since cache miss causes stalling. The clock period decreases. Performance is likely to improve because of locality.

Problem 2: Orthogonal Instruction Encoding

The table below shows the types of base instructions in RICS-V ISA. Assume immediates will be sorted and sign-extended properly in microarchitecture. In this encoding, some bits are used for multiple purposes e.g. bit 7 is used for two purposes: `rd` and `imm`. If we use each bit only for one purpose, how many bits do we need for one instruction? Your encoding must cover all RV32I base instructions except `FENCE`, `ECALL`, and `EBREAK`. Merge opcode and functs and give a new encoding to minimize the bit-width.

Hint 1: we need 20 bits for `imm`.

Hint 2: `shamt` is included in `imm`.



31 27 26 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]		rs1	funct3	rd	opcode	I-type
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12|10:5]	rs2	rs1	funct3	imm[4:1|11]	opcode	B-type
imm[31:12]				rd	opcode	U-type
imm[20|10:1|11|19:12]				rd	opcode	J-type

Solution:

It has 37 instructions so $\lceil \log_2(37) \rceil = 6$ bits are enough to replace opcode and functs. In total, it needs 41 bits = new opcode (6 bits) + rd (5 bits) + rs1 (5 bits) + rs2 (5 bits) + immediate (20 bits).

Problem 3: Memory Address

Imagine you are asked to design a 1 MByte memory with $2N$ rows and N bits per row. Determine N and calculate how many bits you need for the row select signal and the column select signal. The row select signal selects one row, and the column select signal extracts one Byte from the selected row.

Solution:

1 MByte is 2^{23} bits. So,

$$\begin{aligned} 2N^2 &= 2^{23} \\ N^2 &= 2^{22} \\ N &= 2^{11}. \end{aligned}$$

To select one row out of $2N = 2^{12}$ rows, we need 12 bits for the row select signal. To select one Byte from one row, which contains $N/8 = 2^8$ Bytes, we need 8 bits for the column select signal. In total, we have a 20-bit address, which is correct since the memory size is 1 MByte.

Problem 4: Predecoder

Calculate the area of 16-bit decoder for 1-level, 2-level, and 4-level implementations. The 1-level implementation uses one 16-input AND gate for each output. The 2-level implementation has two levels of 4-input AND gates. The 4-level implementation has four levels of 2-input AND gates. Assume the area taken by a N-input AND gate is N, and inverters do not contribute to the area.

Solution:

The area for 1-level implementation is $16 \times 2^{16} = 2^{20}$.
The area for 2-level implementation is $4 \times 4 \times 2^4 + 4 \times 2^{16} = 2^8 + 2^{18}$, where the first level has 4 groups of inputs and 2^4 patterns for each group.
The area for 4-level implementation is $2 \times 8 \times 2^2 + 2 \times 4 \times 2^4 + 2 \times 2 \times 2^8 + 2 \times 2^{16} = 2^6 + 2^7 + 2^{10} + 2^{17}$, where the i-th level has $16/2^i$ groups of inputs and 2^{2^i} patterns for each group.