HOMEWORK 8A

Tuesday, December 8, 2020
10:07 PM

How would 1-bit and 2-bit branch predictors do on the following code?

```
int x = 0;
while (1) {
  int y = 0;
  if (x * y == 0) {
    int z = 0;
    int z = 0;
  }
  else {
    int z = 0;
  }
}
```

Bonus: If I remove the `else` branch, does the accuracy improve?

```
int x = 0;
while (1) {
  int y = 0;
  if (x * y == 0) {
    int z = 0;
    int z = 0;
  }
}
```
For the following constraint graph, construct the bit of the loop:

Loop:
1. LDUR
2. LDMV
3. STUR B
4. ADD
5. CMP
6. B LT
7. SUBH
8. LDMR
9. STUR
10. B LOOP

End:

Assume:

LAR
LAR xo
LAR mem

RAW Sigs
Pull from
Control

Control

Control

RAW x2
(LMAR xo)
Explain what is happening in each stage of the procedure process during the G+ Cycle.

B.LT Loop
CMP #2, X3
ADD #3, X3, X2
STUB X3, #2, X2
Loop: SUB X3, X3, X3, #36
Loop: X2, LDA, #10
Determine the features of the following L1 cache given the following table, which was freshly started. Addresses are aligned to double words and the cache uses the LRU replacement strategy. If a feature cannot be determined, then state so.

<table>
<thead>
<tr>
<th>Address</th>
<th>L1 Hit/Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>Miss</td>
</tr>
<tr>
<td>104</td>
<td>Hit</td>
</tr>
<tr>
<td>96</td>
<td>Hit</td>
</tr>
<tr>
<td>110</td>
<td>Miss</td>
</tr>
<tr>
<td>99</td>
<td>Hit</td>
</tr>
<tr>
<td>107</td>
<td>Hit</td>
</tr>
<tr>
<td>123</td>
<td>Hit</td>
</tr>
<tr>
<td>120</td>
<td>Hit</td>
</tr>
<tr>
<td>111</td>
<td>Miss</td>
</tr>
<tr>
<td>128</td>
<td>Hit</td>
</tr>
<tr>
<td>128</td>
<td>Miss</td>
</tr>
<tr>
<td>128</td>
<td>Miss</td>
</tr>
</tbody>
</table>

- What is the block count? 2
- What is the associativity? 2-way/1FA
- Set associativity? 2 blocks
- FA set association? 2 blocks

2 blocks

DM

 Knocked out 120

≥16

= 12

= 128

= 112

= 96, 111, 112-127, 128-143

Sequential

128-143

112-127

96-111
<table>
<thead>
<tr>
<th>ALU + Load/Store</th>
<th>ALU + Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6: B.LT F00
5: CMP XO, X1
4: SUBS X2, X1, XO
3: ADD X1, XO, X1
2: STUR XO, [X1, #1]
1: LDUR XO, [XO, #0]

Make sure the code schedule is as short as possible.

Schedule the code for a 2-way VLIW with delay slots.
one-word blocks.

associative cache. This cache has 4
miss and conflict miss for a fully
examples of a cold start miss, capacity
Give a list of byte addresses that show
Design a single-cycle CPU that can do “ACC” and “CMOVZ” only:

- **ACC Rd, Rn, Rm**
  - $\text{Reg[Rd]} = \text{Reg[Rd]} + \text{Reg[Rn]} \times \text{Reg[Rm]}$

- **CMOVZ Rd, Rn, Rm**
  - $\text{Reg[Rd]} = (\text{Reg[Rm]} == 0) ? \text{Reg[Rn]} : \text{Reg[Rd]}$

- You do not need to show the control table, but you should clearly show all control signals

- Show the Datapath (including the instruction fetch unit)

- Your machine should be as simple as possible

- You can assume you have available all common blocks of our lab #3 CPU, and a multiplier block
EXIT
ADD X3, X0, X1
B.LT EXIT
SUBS X3, X1, X0
FOR X3, X2, #200
STUR X2, [X3, #200]
LDRR X1, [X3, #200]
ADDI X0, X3, #100

Clock rather than positive or negative branching.
Support forwarding. Write through register the operation (write into register the on negative edge of the for the following code, explain what errors would occur if the lab 4 processor did not
following address accesses: 0, 8, 63, 31, 49, 32, 64, 16
 associative cache with block size of 16 bytes for the
 determine the types of misses on a 64 byte 2-way set