Homework 2 CMPE 110

John Carlyle

January 26, 2014

1. $a = (b + c) * (b * c) + a$

   (a) Stack based

   ```
   push [b]
   push [c]
   add
   push [b]
   push [c]
   mult
   mult
   push [a]
   add
   pop  [a]
   ```

   (b) Accumulator based

   ```
   ld   [b]
   add  [c]
   mult [b]
   mult [c]
   add  [a]
   st   [a]
   ```

   (c) Register-memory based

   ```
   ld   R1, [b]
   add  R2, [c],  R1
   mult R1, [c],  R1
   mult R1, R1, R2
   add  R1, [a],  R1
   st   [a],  R1
   ```

   (d) Load-Store based

   ```
   ld   R1, [a]
   ld   R2, [b]
   ld   R3, [c]
   mult R4, R2, R3
   add  R5, R2, R3
   mult R6, R5, R4
   add  R1, R1, R6
   st   [a],  R1
   ```

2. R1 = src

R2 = pat

R3 = orig

R4 = found

```
mov  R3,   R1
WHILE:
beq  R1,   0,   ENDWHILE
beq  R2,   0,   ENDWHILE
ld   R4,   [R1]
ld   R5,   [R2]
beq  R4,   R5,  TRUTH
jamp ENDIF
mov  R2,   R3
TRUTH:
add  R2,   R2,  4
ENDIF:
jmp  WHILE
ENDWHILE:
ld   R5,   [R2]
bneq R5,   0,   ENDIF2
ld   R4,   0
ENDIF2:
ld   R4,   1
```

3. (a)

A	B	C	D	E	F	G	H	I
0x101	-2	0x103	x	SUB	x	0x111 (7)	0x404	x

(b)

Signal	LD	CALL	SUB
RegDst	1	x	0
RegWrite	1	0	1
ALUSrc	0	x	0
PCSrc	1	0	1
ALUOp	ADD	x	SUB
MemWrite	0	0	0
MemRead	1	0	0
MemToReg	1	0	0
Call	0	1	0

4.

Instruction	MUX D	MUX E
JMP	x	1
BNE foo (taken)	1	0
CALL	x	1
RET	1	0
JMPL R5	1	0
BLE bar (not taken)	0	0