Contents

1 Table of Contents

Designing an ISA to do what? 32K of Block Ram

1.1 Instructions we need

16 core instructions with 4 bit op code (16 registers) 2 extra bits for 'extra flavor', rest are for partitioning (10-bits)

- add
- subtract
- jump
- bit-shifting
- memory access 3 classes (Logical, branching, Memory access)

1.2 Instruction Layout

opcode	flavor	rs	rt
0-3	4-5	6-10	11-15

1.3 Logical Instructions

- Add / Subtract
- Bit shift
- Logical AND OR XOR
- Increment / Decrement

1.4 Branching

- conditional branch reg
- conditional branch label
- jump label
- jump register

1.5 Memory

- leaq
- mov
- push
- pop

1.6 Loop

Loop Code in Psuedo-C

```
sum = 0;
for (i = 0; i < 100; i++)
    sum += a[i];
```

Corresponds to:

```
li %i, 0                ; i = 0
li %sum, 0              ; sum = 0
li %tmp, 0              ; tmp = 0
li %tmp1, 0             ; tmp1 = 0
Loop:
    sll  %tmp, %i, 2    ; tmp = i * 4 (interger)
    add  %tmp, %a       ; tmp = tmp + &a
    lw   %tmp1, %tmp    ; load value @ address tmp into tmp1
    add  %sum, %tmp1    ; sum += tmp
    addi %i, 1          ; i += 1
    li   %cmp, 100      ; load 100 into cmp
    cmp  %i, %cmp       ; compare i < 100
    br   %ZF, Loop      ; branch if Zero Flag set
```