

# Energy Efficient Program Control

6.893 project proposal  
9-26-2000

Ronny Krashinsky and Mike Sung

Program = Computation (*add, shift*) + Data (*load, store*) + **Control** (*branch, jump*)

MIPS RISC:

- Basic block size = 5 to 6 instructions
- `slt, bne (beq)` pair accounts for about 5% of all instructions executed [M. Hampton]

`slt`: register file read (2x32b)  
compare (32b)  
register file write (32b)  
`bne`: register file read (2x32b)  
compare (32b)  
target address calculation  
program counter update

`slt`: register file read (2x32b)  
compare (32b)  
condition register write (1b)  
`bne`: condition register read (1b)  
program counter update

```
for(i=0; i<x; i++)  
  j += i<<1;
```



```
L:  addu $ri $ri 1  
    addu $rj $rj $rs  
    slt  $rc $ri $rx  
    bne  $rc $r0 L  
    sll  $rs $ri 1
```

```
la $btr L  
L:  addu $ri $ri 1  
    addu $rj $rj $rs  
    slt  $c1 $ri $rx  
    b    $c1 (btr)  
    sll  $rs $ri 1
```

Energy savings using *condition registers & branch target register*:

- 1 bit register access for branch condition
- No branch address calculation inside loop
- Possible compact instruction encoding (16 bit)
- Branch resolution in decode stage (1bit check)

Other Ideas:

- Boolean operations can also use 1 bit condition registers
- Alternative control mechanisms:  
    auto-decrement, predication, condition codes
- Program control in alternative architectures:  
    Vector, Data Flow, VLIW

Plan:

- Profile benchmarks (ISA simulator)
- ISA development
- Compiler support (assembly file hacking)
- Micro-architectural implementation/simulation (SyCHOSys)
- Energy simulation (SyCHOSys)
- Analyze and iterate