Register Hierarchy

Roshan Gummattira
Spyros Tsavachidis
Teresa Watkins

April 23, 2002

Register Access Trends

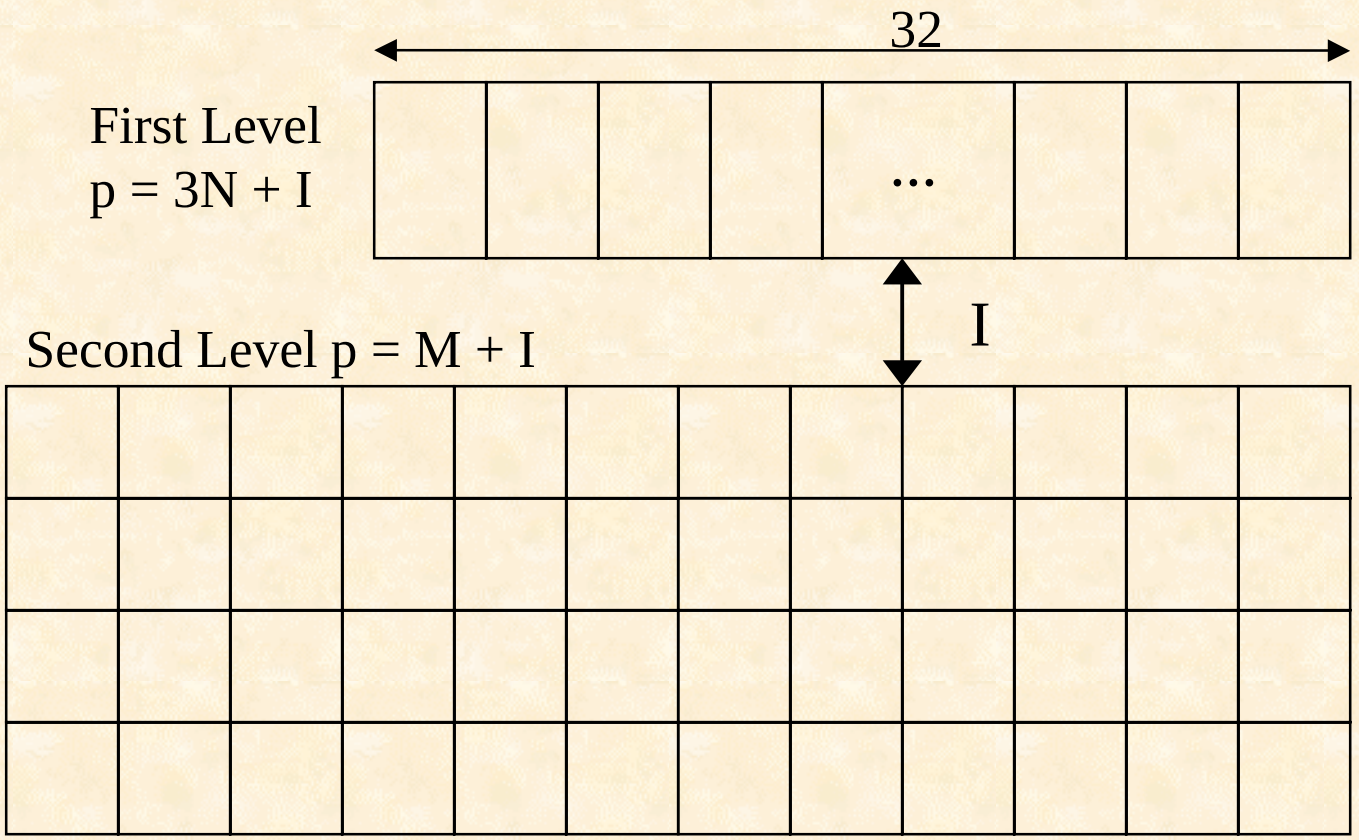
- Access time increases with size of register file
- Register access times do not decrease as fast as clock frequency increases
- Current trends will require smaller register files or multi-cycle register accesses
- Register files with many ports will incur larger access times

Hypothesis

We feel that by breaking up the register file into two levels we can either decrease the size (and therefore latency) of the register file or fit more registers into the same amount of space without introducing new instructions or slowing down the execution pipeline.

Architecture Overview

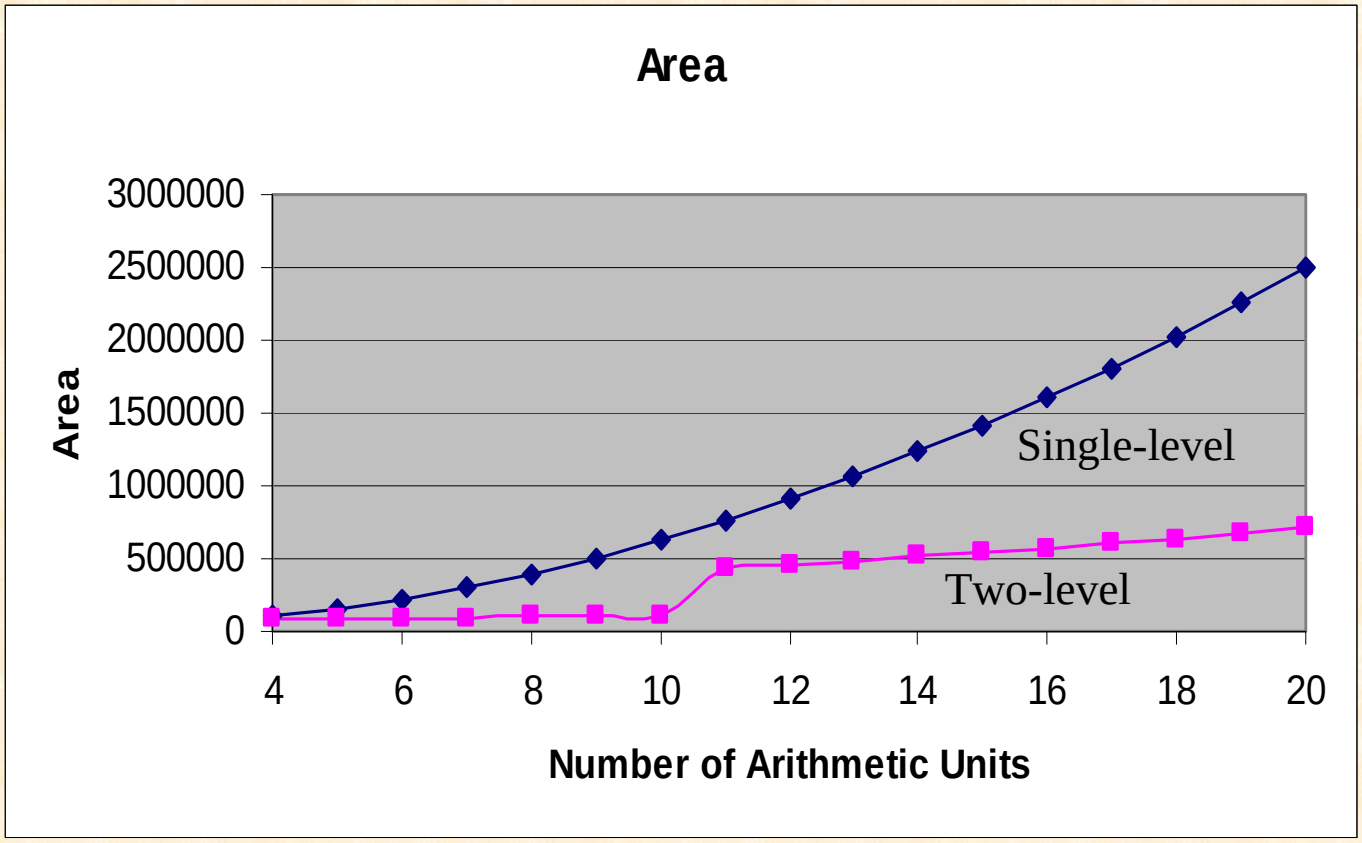
- Top level registers interface only with lower level registers and functional units
- Lower level registers interface only with upper level registers and memory
- Top level registers are a subset of lower level registers
- Limited bandwidth between two levels

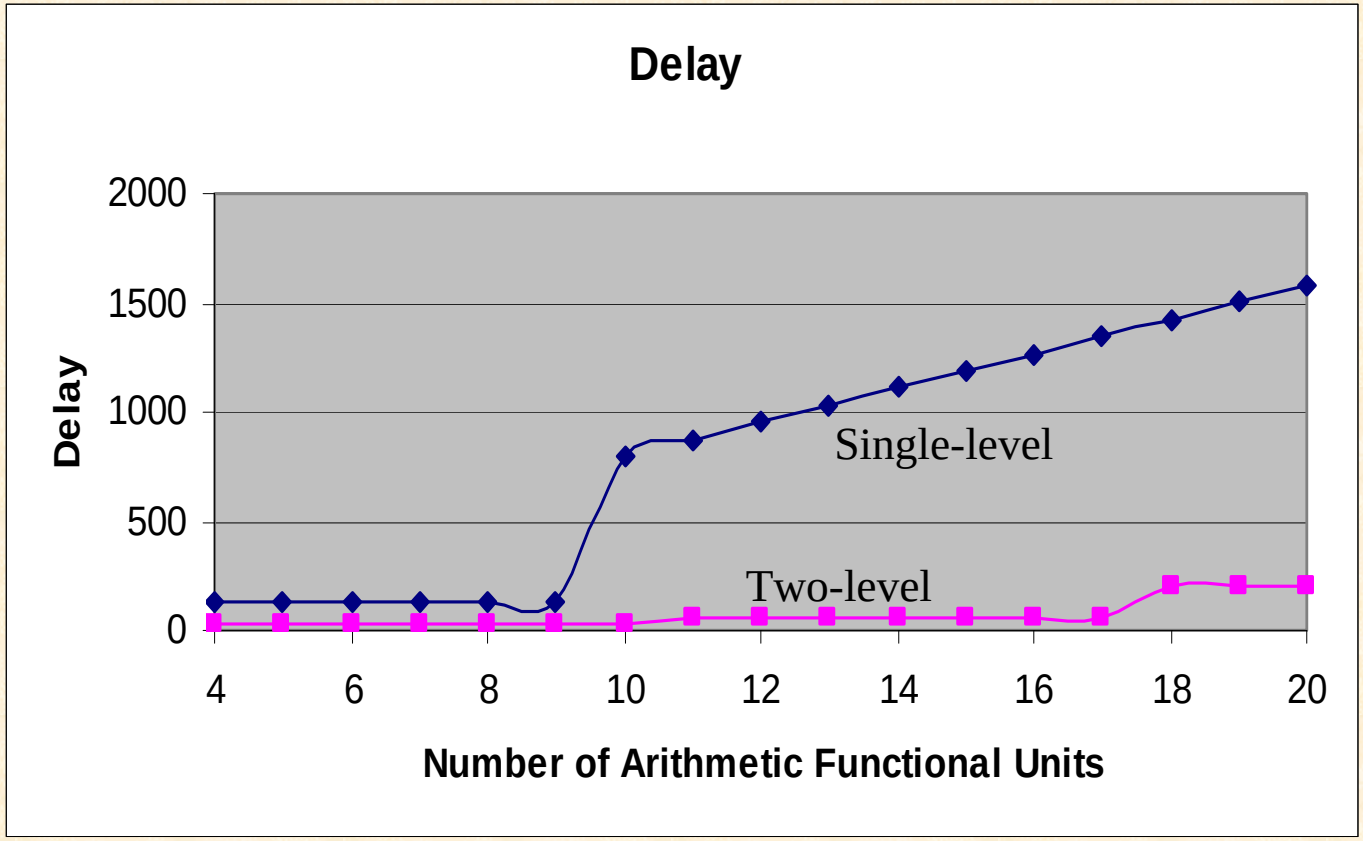

32
First Level
p = 3N + I
...
I
Second Level p = M + I

Register File Model

- M = number of ports to memory
- N = number of functional units
- I = number of ports between levels
- Base case: $p = (M+3)N$
- Two level: $p1 = 3N+I$

  $p2 = I+M$
- Area = Rp^2
- Delay = R for p small, $pR^{1/2}$ for p large


Area
3000000
2500000
2000000
1500000
1000000
500000
0
Area
Single-level
Two-level
4
6
8
10
12
14
16
18
20
Number of Arithmetic Units


Delay
2000
1500
1000
500
0
Delay
Single-level
Two-level
4
6
8
10
12
14
16
18
20
Number of Arithmetic Functional Units

Replacement Policy

- First level a subset of lower level
- Implemented by adding logic to register rename table
- Use a Least Recently Used replacement policy
- Writeback policy for coherence
- Performance highly dependent on register file management

Implementation Issues: Simplescalar

- Simplescalar has fixed number of registers
- Does not have register rename logic
- Proof of concept simulation

  Performance impact of increasing register file latency - unable to test because Simplescalar latencies are based on instructions.

Implementation Issues: RSIM

- RSIM better suited for register file manipulation
- RSIM supports register rename logic
- RSIM supports variable number of registers via number of register windows
- Can only adjust register file access latencies via modifying latencies of all commands

Related Work

- Multiple-banked register file
- Distributed register file
- Compiler-controlled hierarchical register file

Conclusions

- Savings in area because of distribution of ports
- Decrease in delay for the first level
- Performance greatly dependent upon efficiency of replacement policy