MICRO 33

Very Low Power Pipelines using Significance Compression

Ramon Canal, Antonio González and James E. Smith

Universitat Politècnica de Catalunya - Barcelona

University of Wisconsin - Madison

{rcanal, antonio}@ac.upc.es; jes@ece.wisc.edu

UPC

MICRO 33

Motivation

- **Design for low power**
  - Becoming a critical design constraint
  - Types of power consumption
    - Dynamic (predominant source for current technology)
      - Charging/Discharging capacitors
      - Short circuit currents
      - Proportional to the switching activity
    - Static (Increasing dramatically with smaller feature sizes)
      - Leakage current (leaking diodes and transistors)
      - Static currents (design styles as pseudo NMOS)
  - Work to do at technology ... application level
    - This work focused on the architecture part
      - Reduce the dynamic power consumption by reducing the switching activity

UPC

MICRO 33

Motivation

- **Size of the operands**
  - Small values are largely used (that fit in a byte)
    - 00 00 00 01 hex
    - FF FF FF FD hex
  - Large values may have bytes within that are just a sign extension of the previous one.
    - 00 7F 00 01 hex
    - 01 00 00 00 hex
  - Just the significant bytes need to be computed and stored; the rest are redundant

UPC

MICRO 33

Talk Outline

- **Significance compression**
  - Extension bits
  - Basic pipeline
- **Power savings**
  - PC increment
  - Fetch
  - Register file
  - Functional units
  - Memory
  - Writeback
- **Low power pipelines**
- **Conclusions**

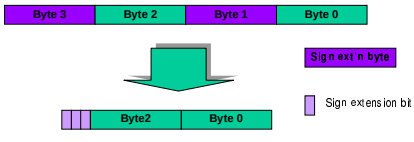
UPC

MICRO 33

Significance Compression

Conventional approach

Significance compression approach



Example: 0x00 7F 00 01 V 101 | -- 7F -- 01

1 extension bit per byte. Byte 0 is always computed so no sign extension needed

UPC

MICRO 33

Significance Compression

- **Size of the operands**
  - values read/written into the register and memory (Avg. Mediabench)

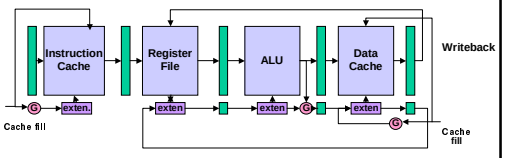
Cases	% of values	Acc	Example (hex)
eees	61.0	61.0	00 00 00 01
eess	13.6	74.6	00 00 01 00
ssss	12.6	87.2	01 01 01 01
esss	7.4	94.6	00 01 01 01
sses	1.8	96.4	01 01 00 01
sess	1.5	97.9	01 00 01 01
eses	1.3	99.2	00 01 00 01
sees	0.8	100	01 00 00 01

  - Overall
    - 61.0% need 8 bits
    - 10.7% need 24 bits
    - 15.7% need 16 bits
    - 12.6% need 32 bits

UPC

MICRO 33

Basic Pipeline

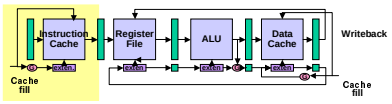


- **Five stage pipeline**
  - Fetch, Decode+Read, Execution, Memory, Writeback
- **Pipeline extended with extension bits**
  - Significance compression in all the stages of the pipeline

UPC

MICRO 33

Power Savings (i)



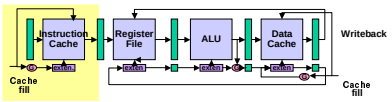
- **PC increment**
  - Compute 1 byte at a time
  - Typically, carry does not propagate between bytes

→ Activity savings 73.3% average

UPC

MICRO 33

Power Savings (ii)



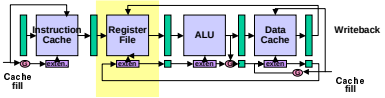
- **Fetch**
  - Compress instructions (from 4 to 3 bytes)
    - Remove fields not used and compress function codes
    - Compress immediates that do not use all the space reserved
  - Most used instructions will use 3 bytes
    - 1 bit extra per instruction to distinguish among compressed/uncompressed instructions

→ Activity savings 18.2% average

UPC

MICRO 33

Power Savings (iii)



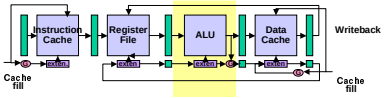
- **Register file (read)**
  - Use significance compression when reading/writing
  - Keep extension bits (3)

→ Activity savings 46.5% average

UPC

MICRO 33

Power Savings (iv)



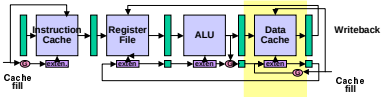
- **ALU**
  - Just compute "significant" bytes
  - Generate extension bits for the result

→ Activity savings 33.2% average

UPC

MICRO 33

Power Savings (v)



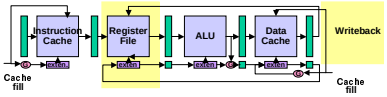
- **Memory (D-cache data)**
  - Just send/receive "significant" bytes
  - Keep extension bits in the cache

→ Activity savings 30.1% average

UPC

MICRO 33

Power Savings (vi)

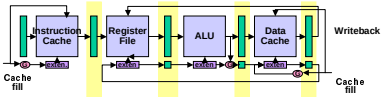


- **Register file (write)**
  - Just write significant bytes
  - Update extension bits
- Activity savings 42.1% average

UPC

MICRO 33

Power Savings (vii)

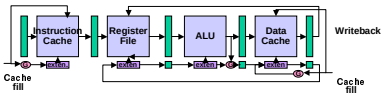


- **Datapath latches**
  - Keep flowing significant bytes
  - Keep extension bits
- Activity savings 42.2% average

UPC

MICRO 33

Power Savings (viii)



- **Summary power savings**

Datapath Width	PC incr	Fetch	RF read	ALU	D-cache data	RF write	Latches
8 bits	73.3%	18.2%	46.5%	33.2%	30.1%	42.1%	42.2%
16 bits	46.7%	18.2%	35.9%	22.1%	23.4%	30.3	34.9%

UPC

MICRO 33

Low Power Pipelines

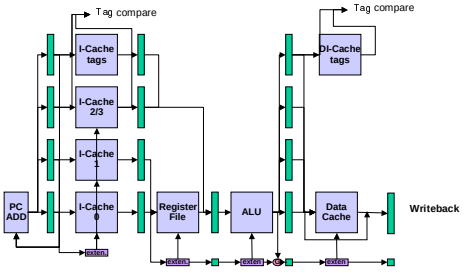
- **Serial**
  - Minimal activity
- **Semi parallel**
  - Throughput balanced
- **Fully-parallel**
  - Complex control
  - Aggressive operand gating
- **Similar activity counts, different performance levels**

UPC

MICRO 33

Low Power Pipelines

- **Byte-serial implementation**

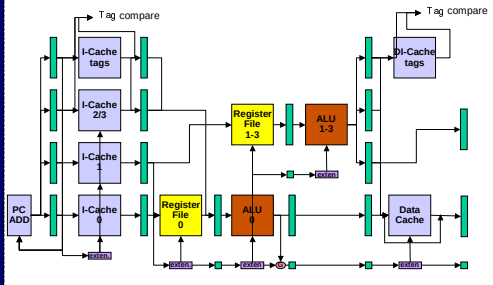


UPC

MICRO 33

Low Power Pipelines

- **Byte semi-parallel implementation**

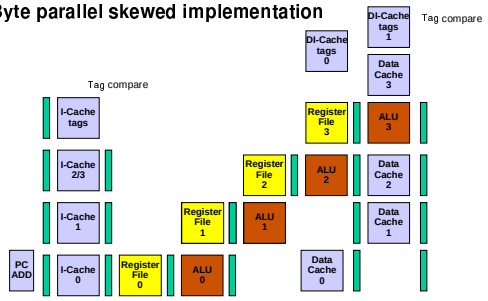


UPC

MICRO 33

Low Power Pipelines

- Byte parallel skewed implementation

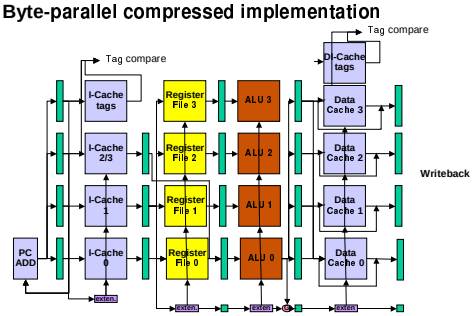


UPC

MICRO 33

Low Power Pipelines

- Byte-parallel compressed implementation

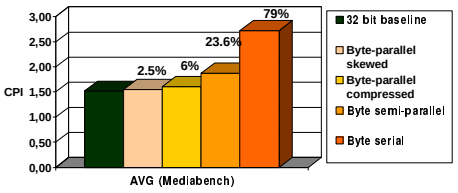


UPC

MICRO 33

Low Power Pipelines

- Byte wide pipelines performance

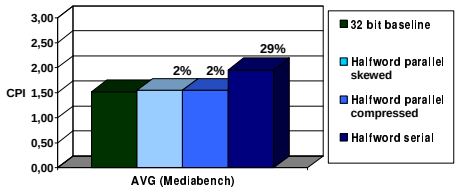


UPC

MICRO 33

Low Power Pipelines

- Halfword wide pipelines performance



UPC

MICRO 33

Summary and Conclusions

- **Significant bytes determine minimal activity**
  - For instructions, addresses and data values
  - This level is 30-40% lower than a conventional 32-bit μarchitecture
- **Proposed pipeline implementations**
  - Several designs proposed
  - Trading off power consumption and performance
    - Similar activity reduction levels
    - Minimal activity achieved by the serial organization
      - Ø Less latches
      - Ø Smaller static consumption (fewer and smaller blocks)
    - Maximum performance achieved by the parallel organizations

UPC