

CS184b: Computer Architecture (Abstractions and Optimizations)

Day 10: April 22, 2005
Statically Compiled ILP
VLIW



Caltech CS184 Spring2005 -- DeHon

Today

- Trace Scheduling
- VLIW uArch
- Evidence for
- What it doesn't address

2

Caltech CS184 Spring2005 -- DeHon

Problem

- Parallelism in Basic Block is limited
 - (recall average branch frequency 7-8 instrs)

3

Caltech CS184 Spring2005 -- DeHon

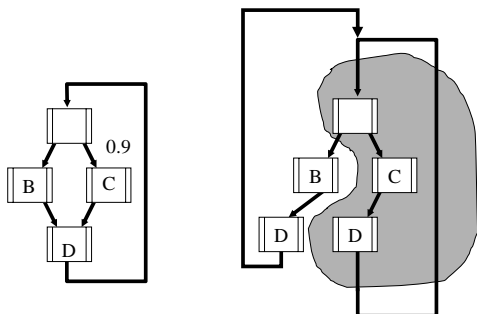
Solution: Trace Scheduling

- Schedule likely sequences of code through branches
 - instrument code
 - capture execution frequency / branch probabilities
 - pick most common path through code
 - schedule as if that happens
 - add “patchup” code to handle *uncommon* case where exit trace
 - repeat for next most common case until done

4

Caltech CS184 Spring2005 -- DeHon

Typical Example



5

Caltech CS184 Spring2005 -- DeHon

Solution Validity

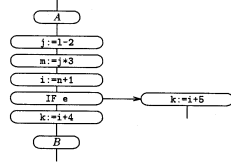
- Recall from Fisher/Predict paper
 - 50-150 instructions/mispredicted branch

6

Caltech CS184 Spring2005 -- DeHon

Trace Example

- Bulldog Fig 4.2

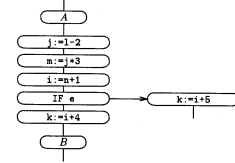


Bulldog: A Compiler for VLIW Architectures
MIT Press 1986
ACM Doctoral Dissertation Award 1985

Caltech CS184 Spring2005 -- DeHon

Trace Example

- Bulldog Fig 4.2



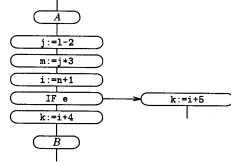
1		IF e1
2	j:=1-2	i:=m+1
3	m:=j+3	k:=i+4

Bulldog: A Compiler for VLIW Architectures
MIT Press 1986
ACM Doctoral Dissertation Award 1985

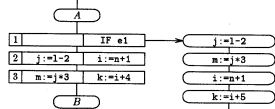
Caltech CS184 Spring2005 -- DeHon

Trace Example

- Bulldog Fig 4.2



1		IF e1
2	j:=1-2	i:=m+1
3	m:=j+3	k:=i+4

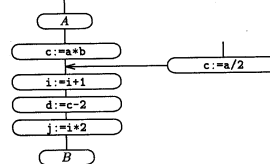


Bulldog: A Compiler for VLIW Architectures
MIT Press 1986
ACM Doctoral Dissertation Award 1985

Caltech CS184 Spring2005 -- DeHon

Trace Join Example

Bulldog p61



the code generator produced the following schedule:

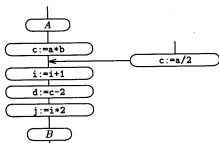
1	c:=a*b	i:=i+1
2	d:=c-2	j:=i+2

Caltech CS184 Spring2005 -- DeHon

10

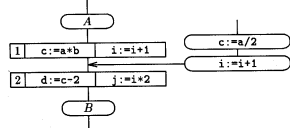
Trace Join Example

Bulldog p61-62



the code generator produced the

1	c:=a*b	i:=i+1
2	d:=c-2	j:=i+2

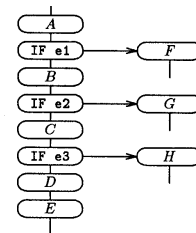


Caltech CS184 Spring2005 -- DeHon

11

Trace Multi-Branch Example

Bulldog p69

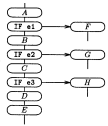


1	A			
2	E	IF e1	IF e2	IF e3
3	B	C	D	

Caltech CS184 Spring2005 -- DeHon

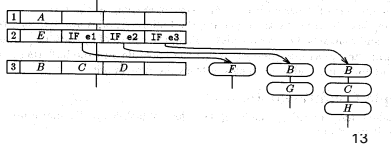
12

Trace Multi-Branch Example



Bulldog p69-70

1	A			
2	B	IF e1	IF e2	IF e3
3	B	C	D	



Caltech CS184 Spring2005 -- DeHon

13

Trace Advantage

- Avoid fragmentation
 - can't fill issue slots because broken by branches
- Expose more parallelism
 - concurrently run things on different sides of branches
 - allow more global code motion (across branches)

Caltech CS184 Spring2005 -- DeHon

14

Loops

- **Problem:** loops introduce (conditional) branches
 - Breaks up code to schedule
 - Adds overhead for testing
 - Maybe limited parallelism in single loop body

Caltech CS184 Spring2005 -- DeHon

15

Loop Unrolling

- **Solution:** unroll the loop
 - Create larger basic block and trace-schedule
 - More stuff to work with
 - Loop less frequently
 - Amortize out loop control overhead
 - Common case will be many iterations of loop

Caltech CS184 Spring2005 -- DeHon

16

Example

- $i:=1$
- LOOP
 - IF $i>n$ THEN EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$

Caltech CS184 Spring2005 -- DeHon

17

Example Cont.

- $i:=1$
- LOOP
 - IF $i>n$ THEN EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
- $i:=1$
- LOOP
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$

Caltech CS184 Spring2005 -- DeHon

18

Example Cont.

- $i:=1$
- LOOP
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
 - If $i>n$ then EXITLOOP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+1$
- Trace Schedule, Rename ...
- $i:=1, j=2, k=3$
- LOOP
 - If $i+3>n$ then CLEANUP
 - $A[i]:=b[i]+c[i]$
 - $i:=i+3$
 - $A[j]:=b[j]+c[j]$
 - $j:=j+3$
 - $A[k]:=b[k]+c[k]$
 - $k:=k+3$
- CLEANUP
 - ...

Caltech CS184 Spring2005 -- DeHon

19

Machine

- Single PC/thread of control
- Wide instructions
- Branching
- Register File
- Memory Banking

Caltech CS184 Spring2005 -- DeHon

20

Branching

- Allow multiple branches per "Instruction"
 - n-way branch
- N-tests + 1 fall-through
 - order in trace order
 - take first to succeed
- Encoding
 - single base address
 - branch to base+i
 - i is test which succeeded

Caltech CS184 Spring2005 -- DeHon

21

Split Register File

- Each cluster has own RF
 - (register bank)
 - can have limited read/write bw
- Limited networking between clusters
 - explicit moves between clusters when results needed elsewhere

Caltech CS184 Spring2005 -- DeHon

22

Memory Banks

- Separate Memory Banks
 - dispatch set of non-conflicting loads/stores, each to separate memory banks
 - trick is can compiler determine non-conflict
 - (do layout to avoid conflicts)
 - has to know won't conflict (for VLIW timing)

Caltech CS184 Spring2005 -- DeHon

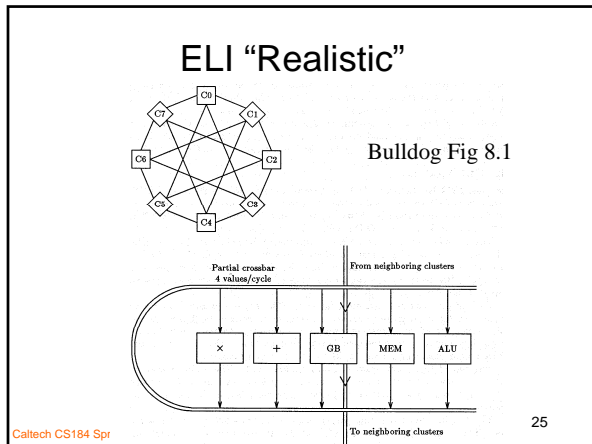
23

Memory Banks

- Avoid single memory bottleneck
- Avoid having to build n-ported memory
- Can make likelihood of conflict small
- Costs for crossbar between memory and consumers
- Arbitration required if can't statically schedule access pattern
- Hotspots/poor bank allocation can degrade performance

Caltech CS184 Spring2005 -- DeHon

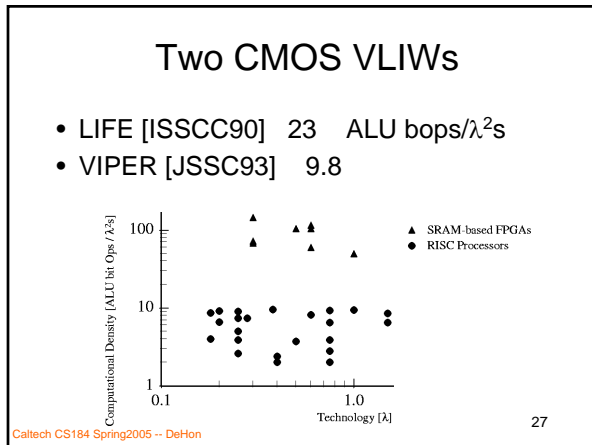
24



Ellis Results

Program	Speed-up		Input size		Unrolling	
	Ideal	Real.	Ideal	Real.	Ideal	Real.
MATMUL	25.5	7.4	50 ²	50 ²	32	64
FFT	48.3	6.9	1024	1024	16	16
SOLVE	18.9	6.2	128 ²	128 ²	16	16/8
SVD	16.2	5.4	30 ²	128 ²	16/2	32/16
TRID1	2.7	.9	4096	4096	16	8
TRID2	3.8	1.2	4096	4096	16	8
TRID4	33.3	7.0	4096	4096	16/4	16/2
EOS	8.3	2.3	64 ²	64 ²	16	8
NEWZR	19.8	7.6	60 ²	49 ²	8	16
QK61	10.2	4.5	-	-	2	4
QUANC8	8.1	-	-	-	4	-
ZEROIN	3.5	-	-	-	4	-

Caltech CS184 Spring2005 -- DeHon Bulldog p242 26



- ### Modern VLIW Examples
- Trimedia
 - TI 6x series
 - Transmeta

 - Academic descendants
 - M-Machine
 - RAW
- Caltech CS184 Spring2005 -- DeHon 28

- ### What can/can't it do?
- Multiple Issue?
 - Renaming?
 - Branch prediction?
 - static
 - dynamic
 - Tolerate variable latency?
 - memory
 - functional units
- Caltech CS184 Spring2005 -- DeHon 29

- ### Scaling
- Issue
 - Bypass
 - Register File
 - N-way branch
 - Memory Banking
 - RF-RF datapath
- Caltech CS184 Spring2005 -- DeHon 30

Scaling

- Linear Scaling
 - Issue
 - Bypass (only within cluster)
 - Register File (separate per cluster)
- Super linear
 - Memory Banking [(clusters)² ?]
 - RF-RF datapath ?
 - Unclear from small examples (and didn't study)

Caltech CS184 Spring2005 -- DeHon

31

Scaling: N-way branch?

- Probably want to scale up branching with clusters (VLIW length)
- Use parallel prefix computation
 - depth goes as log(N)
 - area can be linear

Caltech CS184 Spring2005 -- DeHon

32

Scaling: Thoughts

- W/ on-chip memory
 - banks local to clusters (distributed memory)
 - can schedule operations on clusters close to memory? → MIT RAW
 - Communicate data among clusters (like RF to RF transfers) if need non-local
 - How much interconnect needed?
 - What's the locality of data communication?
 - Recall interconnect richness study from last term

Caltech CS184 Spring2005 -- DeHon

33

“Weaknesses”

- Binary Compatibility
 - lack thereof
- No “Architecture”
- Exceptions

Caltech CS184 Spring2005 -- DeHon

34

VLIW Roundup

- Exploit ILP
- w/out all the hardware complexity and cost
- Relegate even more interesting stuff to the compiler (REMISC?)
- ...but no binary compatibility path
 - maybe not important w/ JIT+Binary Trans.

Caltech CS184 Spring2005 -- DeHon

35

Admin

- ???

Caltech CS184 Spring2005 -- DeHon

36

Big Ideas

- Get better packing/performance scheduling large blocks
- Common case
- Feedback
 - (future like past)
 - discover common case
- Binding Time hoisting
 - Don't do at runtime what you can do at compile time
- Stable abstraction

Caltech CS184 Spring2005 -- DeHon

37