Appendix C
Architectural Alternatives: the Number of Addresses

Arthur B. Maccabe
Department of Computer Science
The University of New Mexico

Overview

Slide 1

- Basic characterizations
  - counting memory references
  - 3 explicit addresses
  - 2 explicit addresses
  - 1 explicit address
  - 0 explicit addresses
- Adding registers to a
  - 3-address machine
  - 2-address machine
  - 1-address machine
  - 0-address machine
  - load/store machine

Preliminaries

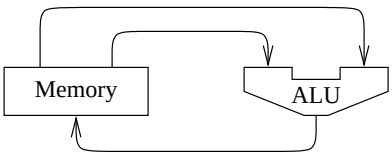
Slide 2

- All operands in memory: operand $\equiv$ memory address
- Assume that each address is stored in an instruction extension word
- Memory addresses use the notation "[*address*]"
- Accounting:
  - instruction references
  - data references

3-Address Architecture

Slide 3

- All three addresses are explicit
- Data paths



An example

$$a = a * b + c * d * e$$

a	100
b	108
c	116
d	124
e	132

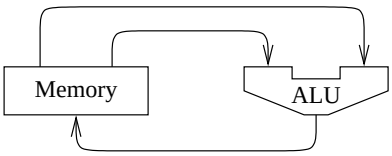
Slide 4

mulx	[100], [108], [100]	! $a * b \rightarrow a$	4 instr. refs	3 data refs
mulx	[116], [124], [200]	! $c * d \rightarrow t$	4 instr. refs	3 data refs
mulx	[200], [132], [200]	! $t * e \rightarrow t$	4 instr. refs	3 data refs
add	[100], [200], [100]	! $a + t \rightarrow a$	4 instr. refs	3 data refs
			16 instr. refs	12 data refs

2-Address Architecture

Slide 5

- 2 explicit addresses
  - one address is a simple source
  - the other is a source and destination
- Data paths



An Example

$$a = a * b + c * d * e$$

a	100
b	108
c	116
d	124
e	132

Slide 6

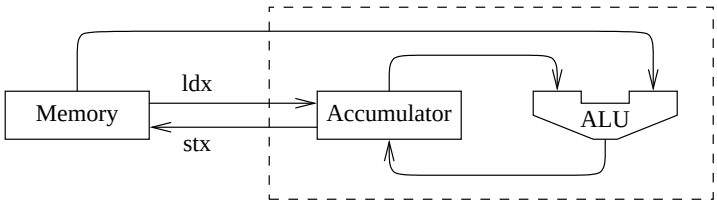
mulx	[108], [100]	! $b * a \rightarrow a$	3 instr. refs	3 data refs
mov	[116], [200]	! $c \rightarrow t$	3 instr. refs	2 data refs
mulx	[124], [200]	! $d * t \rightarrow t$	3 instr. refs	3 data refs
mulx	[132], [200]	! $e * t \rightarrow t$	3 instr. refs	3 data refs
add	[200], [100]	! $t + a \rightarrow a$	3 instr. refs	3 data refs
			15 instr. refs	14 data refs

1-Address Architecture

- One explicit address
  - *accumulator* is the second source and the destination
  - *ldx* load the accumulator
  - *stx* store the accumulator

Slide 7

- Data paths



Slide 8

An Example

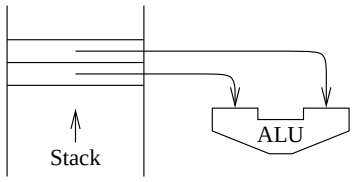
$$a = a * b + c * d * e$$

ldx	[100]	! acc = a	2 instr. refs	1 data ref
mulx	[108]	! acc = acc $* b$	2 instr. refs	1 data ref
stx	[100]	! a = acc	2 instr. refs	1 data ref
ldx	[116]	! acc = c	2 instr. refs	1 data ref
mulx	[124]	! acc = acc $* d$	2 instr. refs	1 data ref
mulx	[132]	! acc = acc $* e$	2 instr. refs	1 data ref
add	[100]	! acc = acc $+ a$	2 instr. refs	1 data ref
stx	[100]	! a = acc	2 instr. refs	1 data ref
			16 instr. refs	8 data refs

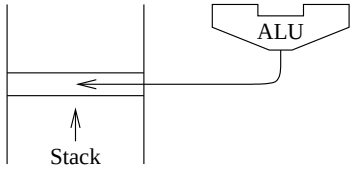
Slide 9

0-Address Architecture

- No explicit addresses
  - the *stack* provides sources and destination
  - *pushx* and *popx*
- Fetching the operands



- Saving the result



An Example

$$a = a * b + c * d * e$$

Slide 10

pushx	[100]	! $\langle a \rangle$	2 instr. refs	1 data ref
pushx	[108]	! $\langle a \rangle \langle b \rangle$	2 instr. refs	1 data ref
mulx		! $\langle a * b \rangle$	1 instr. ref	
pushx	[116]	! $\langle a * b \rangle \langle c \rangle$	2 instr. refs	1 data ref
pushx	[124]	! $\langle a * b \rangle \langle c \rangle \langle d \rangle$	2 instr. refs	1 data ref
mulx		! $\langle a * b \rangle \langle c * d \rangle$	1 instr. ref	
pushx	[132]	! $\langle a * b \rangle \langle c * d \rangle \langle e \rangle$	2 instr. refs	1 data ref
mulx		! $\langle a * b \rangle \langle c * d * e \rangle$	1 instr. ref	
add		! $\langle a * b + c * d * e \rangle$	1 instr. ref	
popx	[100]		2 instr. refs	1 data ref
			16 instr. refs	6 data refs

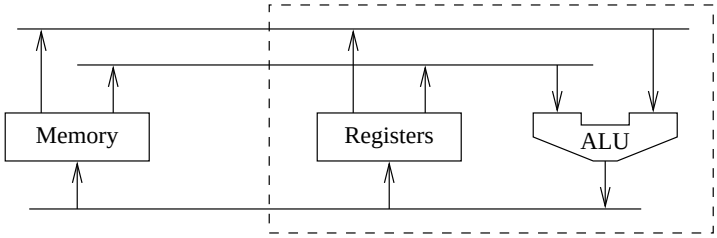
Registers

Slide 11

- Reduce data references (intermediate results)
- Reduce instruction references (small addresses)

Slide 12

Data Paths for 2- and 3-Address



Slide 13

3-Address Example

mulx	[100], [108], %r2	! $a * b \rightarrow$ %r2	3 instr. refs	2 data refs
mulx	[116], [124], %r3	! $c * d \rightarrow$ %r3	3 instr. refs	2 data refs
mulx	[132], %r3, %r3	! $e *$ %r3 $\rightarrow$ %r3	2 instr. refs.	1 data ref
add	%r3, %r2, [100]	! %r3 $+$ %r2 $\rightarrow a$	2 instr. refs	1 data ref
			10 instr. refs	6 data refs

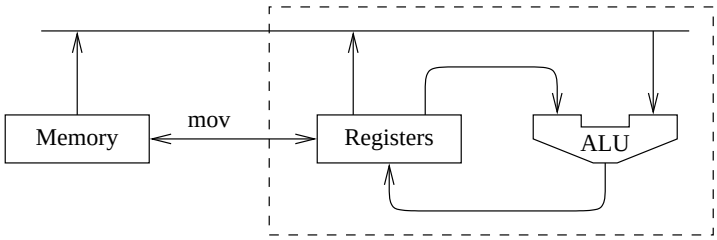
Slide 14

2-Address Example

mulx	[108], [100]	! $b * a \rightarrow a$	3 instr. refs	3 data refs
mov	[116], %r2	! $c \rightarrow$ %r2	2 instr. refs	1 data ref
mulx	[124], %r2	! $d *$ %r2 $\rightarrow$ %r2	2 instr. refs	1 data refs
mulx	[132], %r2	! $e *$ %r2 $\rightarrow$ %r2	2 instr. refs	1 data ref
add	%r2, [100]	! %r2 $+ a \rightarrow a$	2 instr. refs	2 data refs
			11 instr. refs	8 data refs

Slide 15

1 1/2-Address



Slide 16

1 1/2-Address Example

mov	[100], %r2	! $a \rightarrow$ %r2	2 instr. refs	1 data ref
mulx	[108], %r2	! $b *$ %r2 $\rightarrow$ %r2	2 instr. refs	1 data ref
mov	[116], %r3	! $c \rightarrow$ %r3	2 instr. refs	1 data ref
mulx	[124], %r3	! $d *$ %r3 $\rightarrow$ %r3	2 instr. refs	1 data ref
mulx	[132], %r3	! $e *$ %r3 $\rightarrow$ %r3	2 instr. refs	1 data ref
add	%r3, %r2	! %r3 $+$ %r2 $\rightarrow$ %r2	1 instr. ref	
mov	%r2, [100]	! %r2 $\rightarrow a$	2 instr. refs	1 data ref
			13 instr. refs	6 data refs

Slide 17

1-Address Example

ldx	[100]	! acc $= a$	2 instr. refs	1 data ref
mulx	[108]	! acc $=$ acc $* b$	2 instr. refs	1 data ref
stx	%r2	! %r2 $=$ acc	1 instr. ref	
ldx	[116]	! acc $= c$	2 instr. refs	1 data ref
mulx	[124]	! acc $=$ acc $* d$	2 instr. refs	1 data ref
mulx	[132]	! acc $=$ acc $* e$	2 instr. refs	1 data ref
add	%r2	! acc $=$ acc $+$ %r2	1 instr. ref	
stx	[100]	! $a =$ acc	2 instr. refs	1 data ref
			14 instr. refs	6 data refs

Load/Store Example

Slide 18

ldx	[100], %r2	! *load a*	2 instr	1 data
ldx	[108], %r3	! *load b*	2 instr	1 data
mulx	%r3, %r2, %r2	! $a * b \rightarrow$ %r2	1 instr	
ldx	[116], %r4	! *load c*	2 instr	1 data
ldx	[124], %r5	! *load d*	2 instr	1 data
mulx	%r4, %r5, %r4	! $c * d \rightarrow$ %r4	1 instr	
ldx	[132], %r6	! *load e*	2 instr	1 data
mulx	%r4, %r6, %r4	! $c * d * e \rightarrow$ %r4	1 instr	
add	%r4, %r2, %r2	! $c * d * e + a * b \rightarrow$ %r2	1 instr	
stx	%r2, [100]		2 instr	1 data
			16 instr	6 data