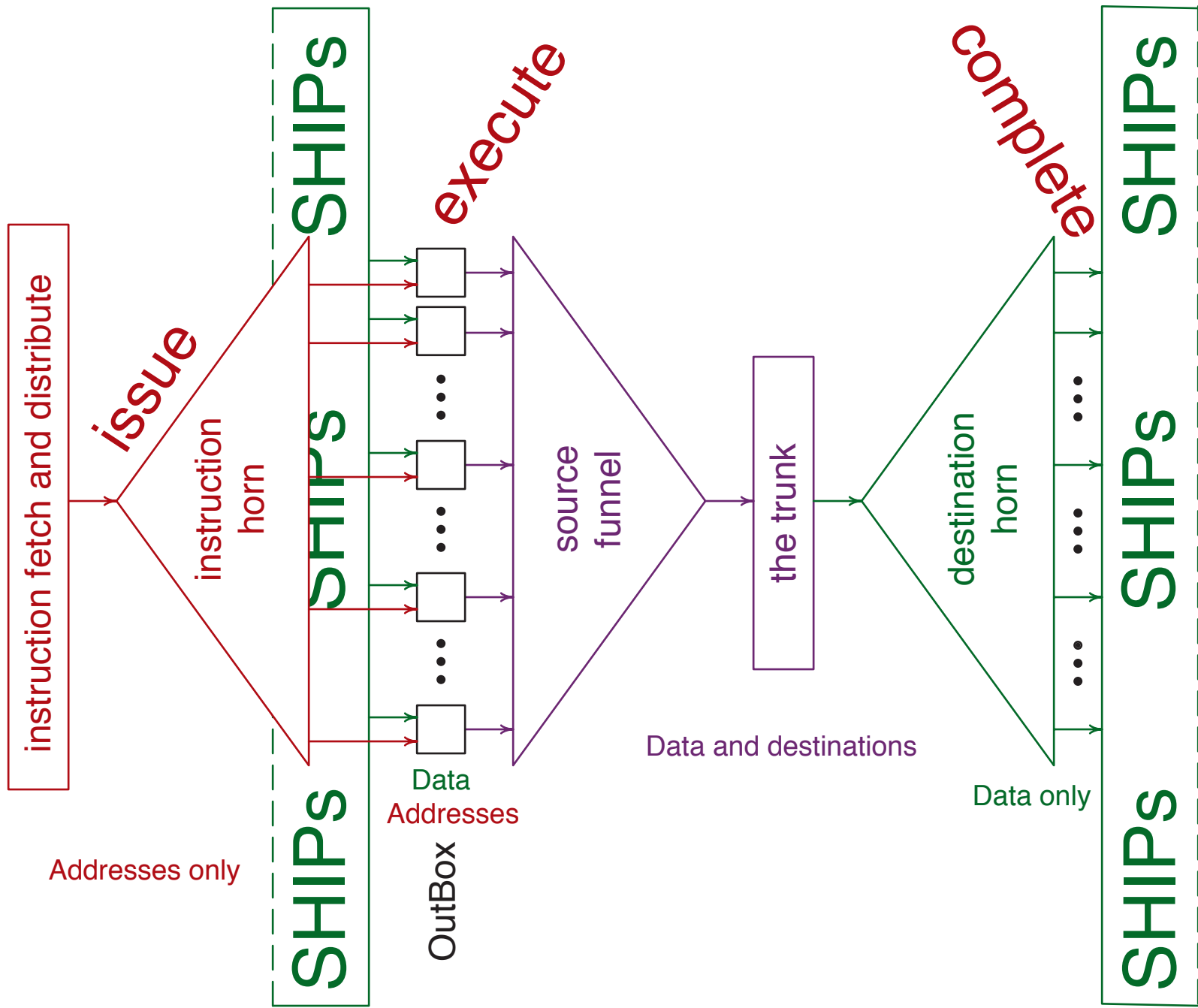


Some SHIPs

Ivan E. Sutherland

September 20, 2006

research.cs.berkeley.edu/fleet



Concurrent MOVEs

When MOVEs execute

- MOVEs execute when data are available
- Each MOVE waits as long as required
- All waiting MOVEs are concurrent

Code Bags

- Limited sequence inside bag
- Zero or more successor bags
- Each bag must fetch successor(s)
- Each BAG's code enters
the POOL concurrently

Sequence guarantees

Source sequence

MOVEs with same source

execute in issue sequence

Identical instructions complete in

execute sequence

Bag is a bag of lists with same source

Code bag sequence

All MOVEs in bag issue before

any from the "next" bag.

Two MOVE types

MOVE vs COPY

MOVE drains source data

COPY preserves source data

Plus

MOVEs with same source

execute in issue sequence

COPY . . . MOVE duplicates

Data Types

Token (zero bits)

Boolean (one bit)

Word (N bits)

Code Bag Descriptor (? bits)

Record = 1 to 8 words

Ordered

Knows own length

Atomic record MOVE

Source variants

COPY record - preserves it

MOVE record - drains it

COPY 1st word - preserves it

MOVE 1st word - record shorter

Destination variants

Partial - record remains open

Final - record sealed

$A_w \Rightarrow B_p; A_w \Rightarrow B_f; A \Rightarrow C;$

Simple SHIPs

Single address

Bit bucket

Constants: ZERO, TRUE, TOKEN

Random number

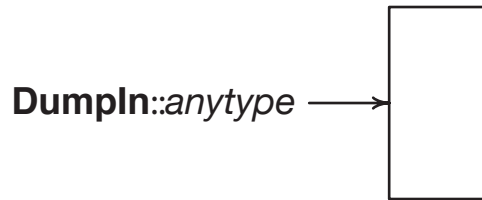
Record store

One or more records

First In First Out (FIFO)

Capacity of FIFO

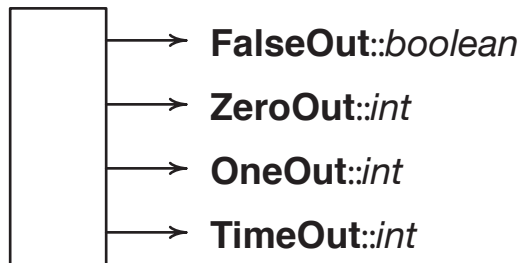
Simple SHIPs



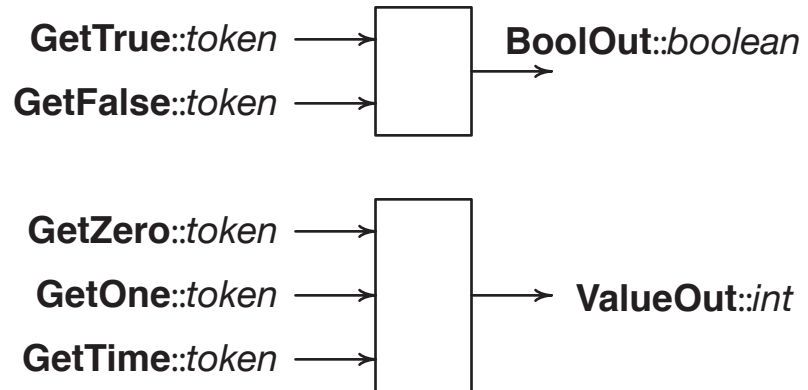
A: Bit Bucket



B: TokenSource

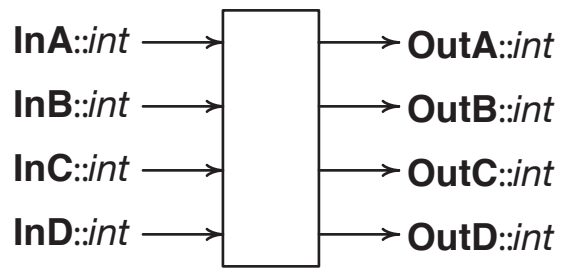


C: Constants

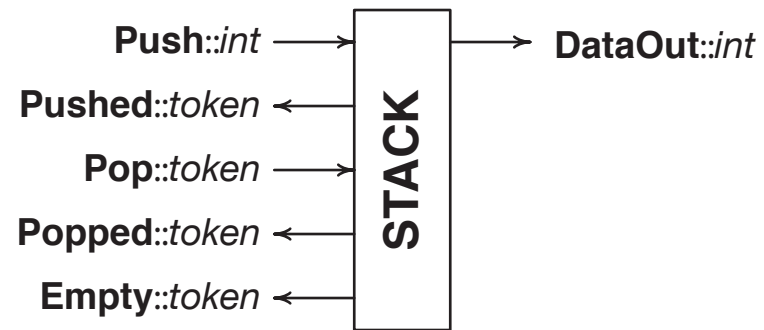


D: Get Constants

Simple SHIPs cont'd



A: Barrier SHIP

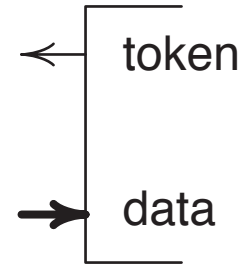


B: Stack SHIP

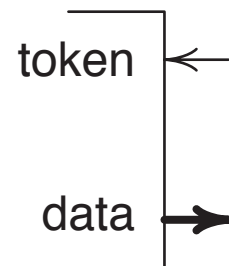
Interfaces

A token appears here after each data input, but only when the interface is ready for further input.

Data arriving here may never be overwritten.
Record may fill the interface, blocking further input until the interface is ready.



Pipeline receiver interface



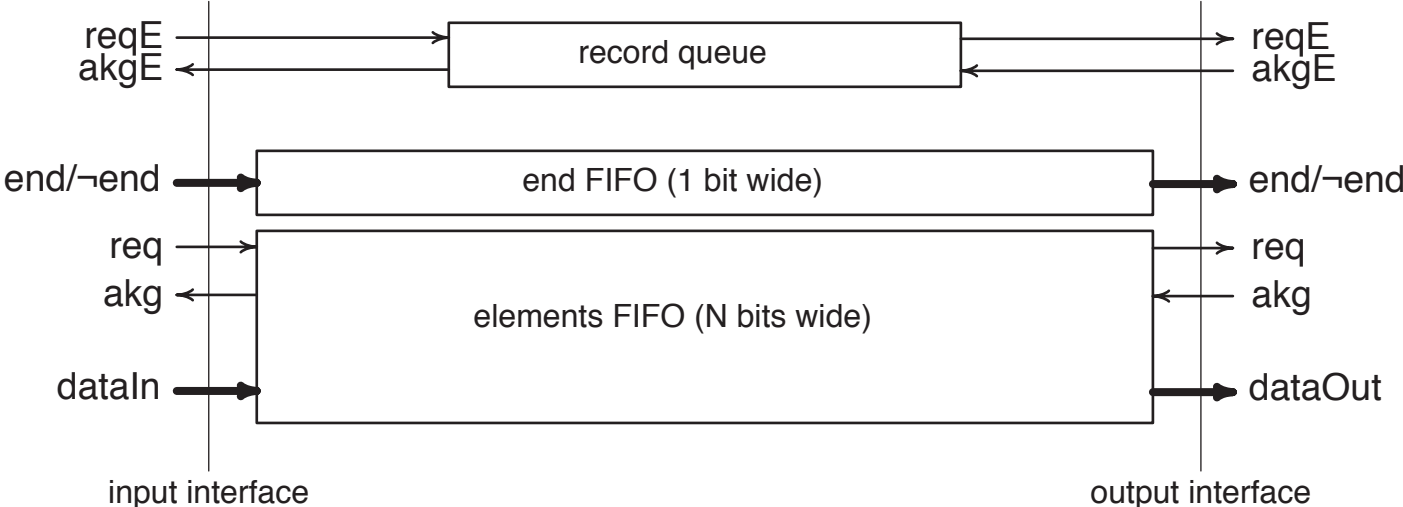
This destination must receive a fresh token for each new output data value.

A new record appears here, if available, only after the previous record is drained and a fresh token has arrived.
Move instructions may copy or drain data values.

Pipeline sender interface

Connect with two MOVEs

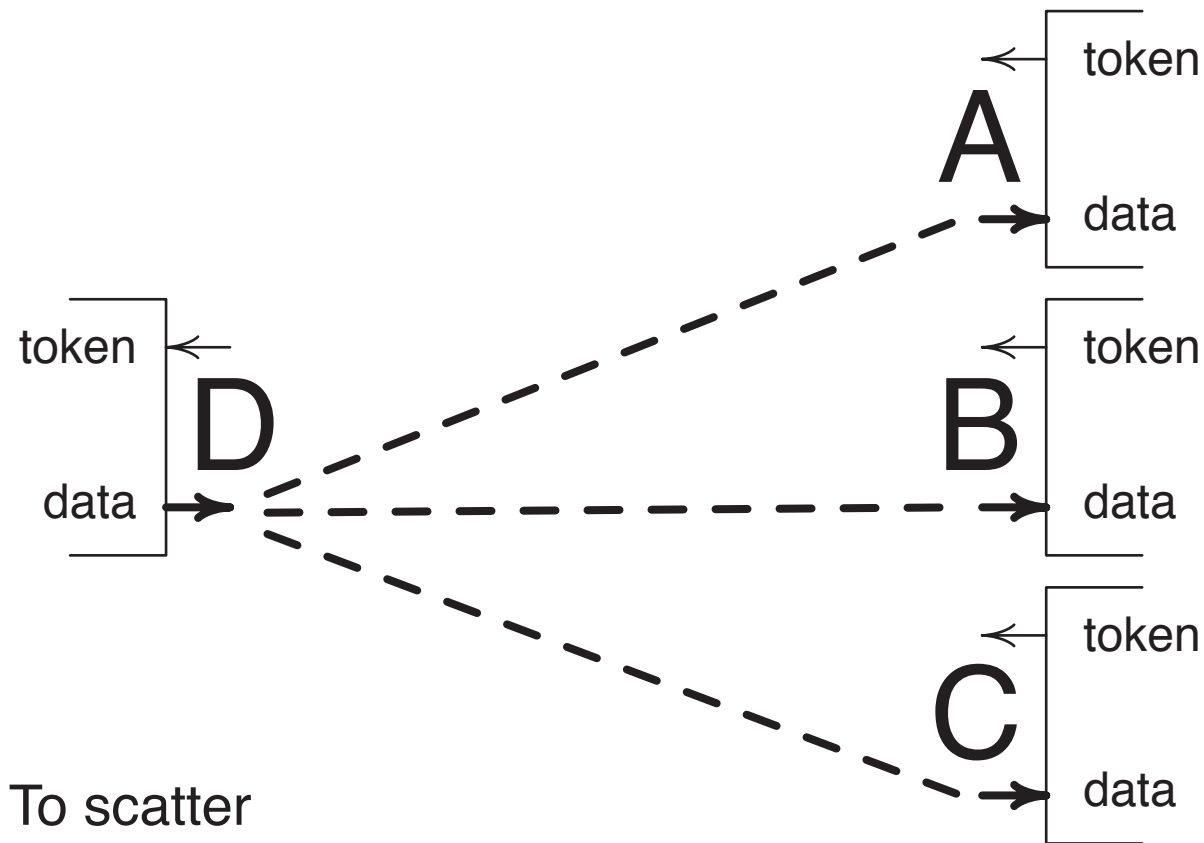
Record Store



			empty	empty	empty	empty	empty	empty	record	record
	empty	empty	1 = end	0	1 = end	0	0	0		
	empty	empty	data6	data5	data4	data3	data2	data1		

Holding a four-word record and a two-word record.

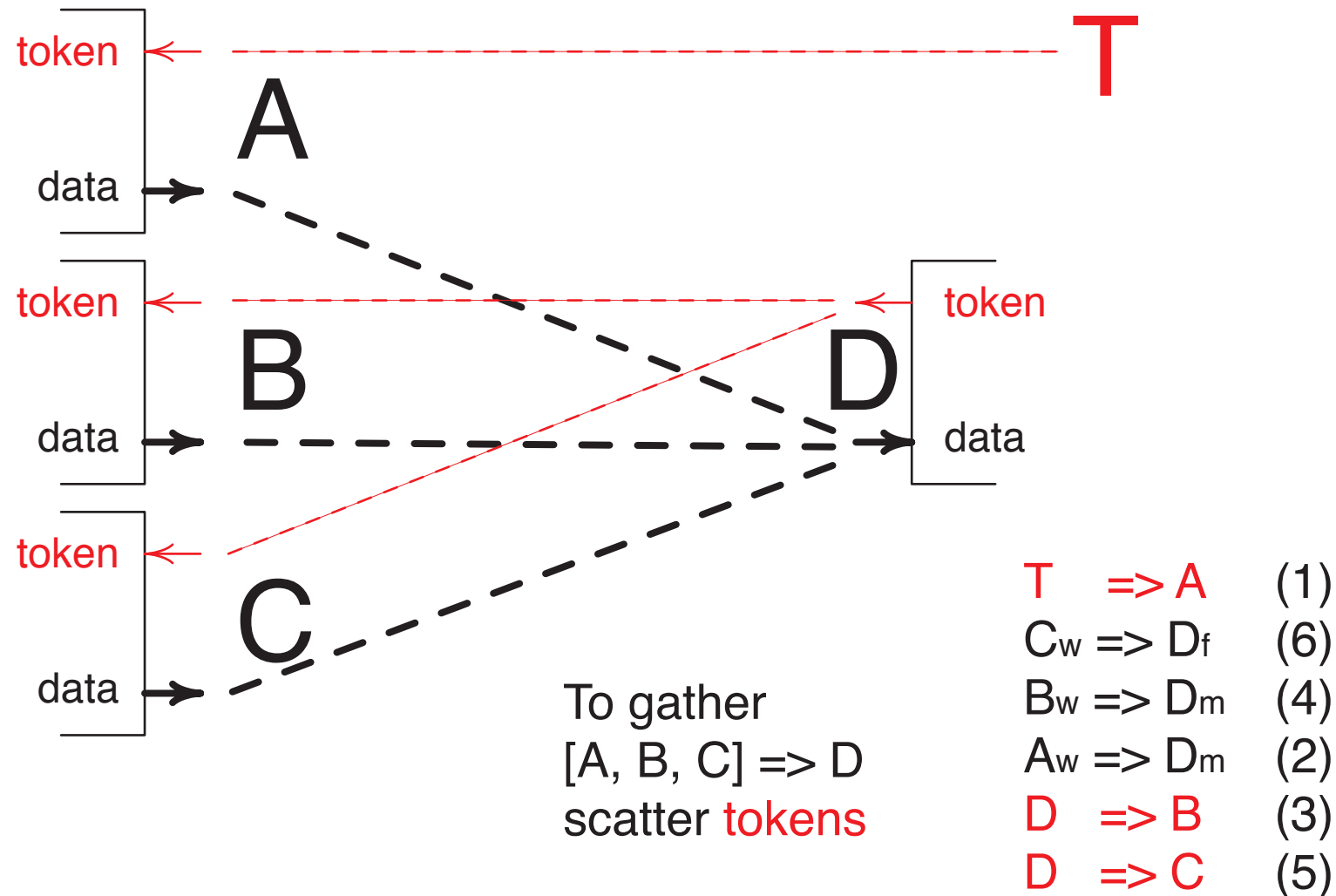
Scatter via source sequence



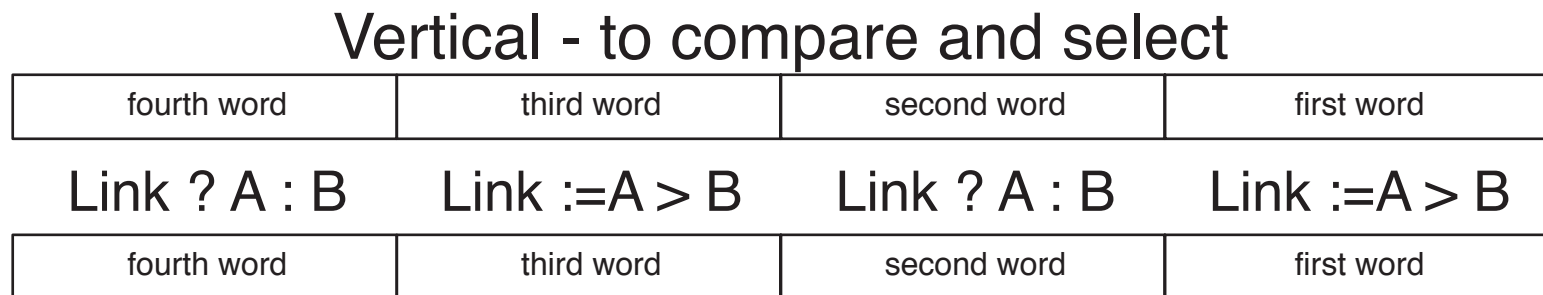
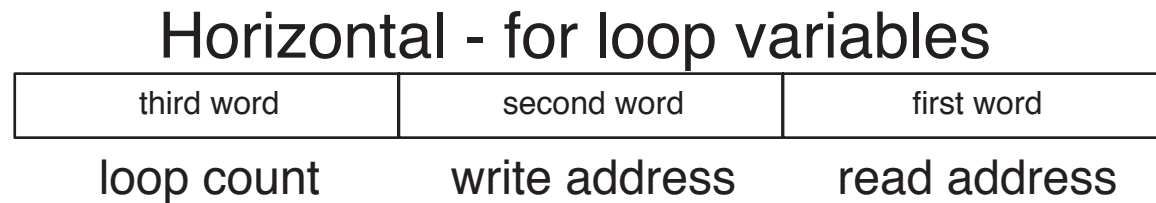
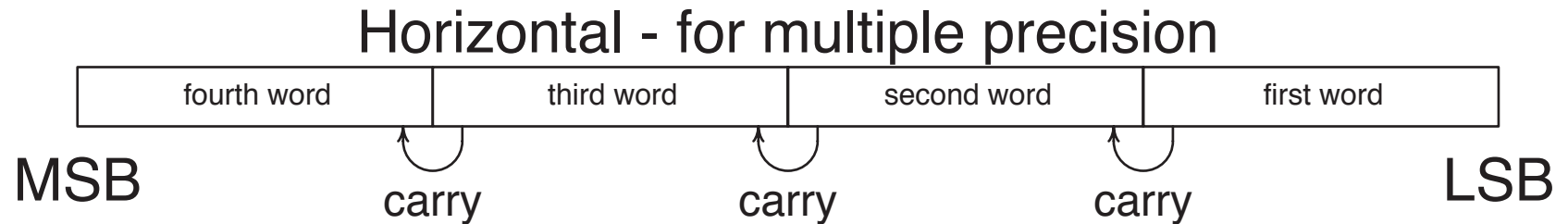
To scatter
[D1, D2, D3] => A, B, C
use source sequence

- $D_w \Rightarrow A_f$ (1)
- $D_w \Rightarrow B_f$ (2)
- $D_w \Rightarrow C_f$ (3)

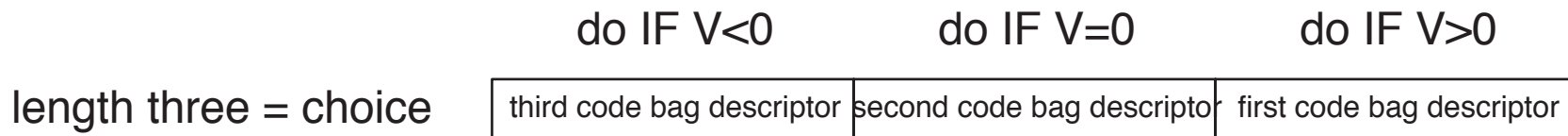
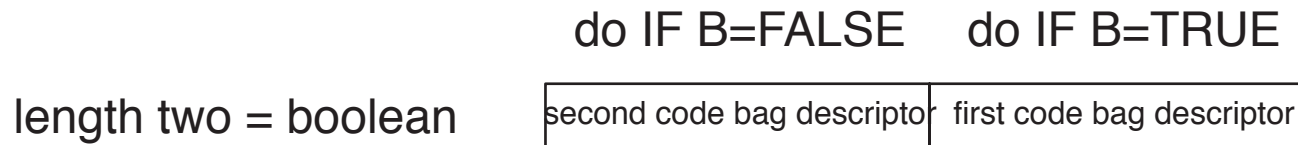
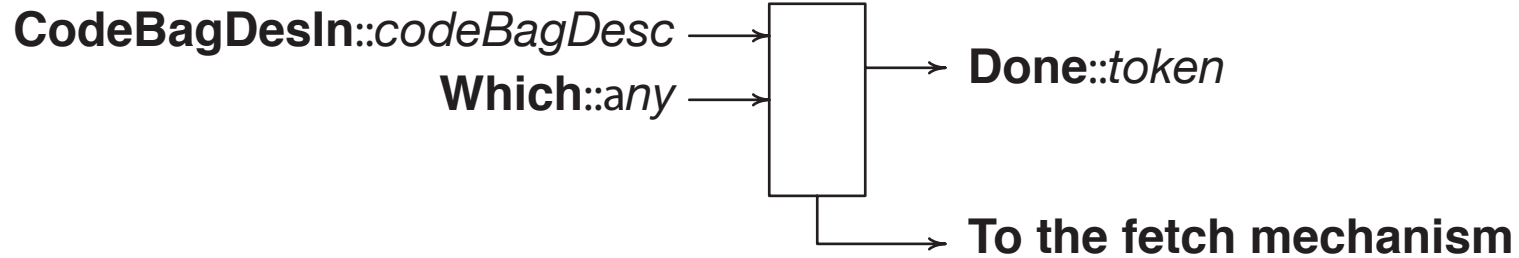
Scatter tokens to gather data



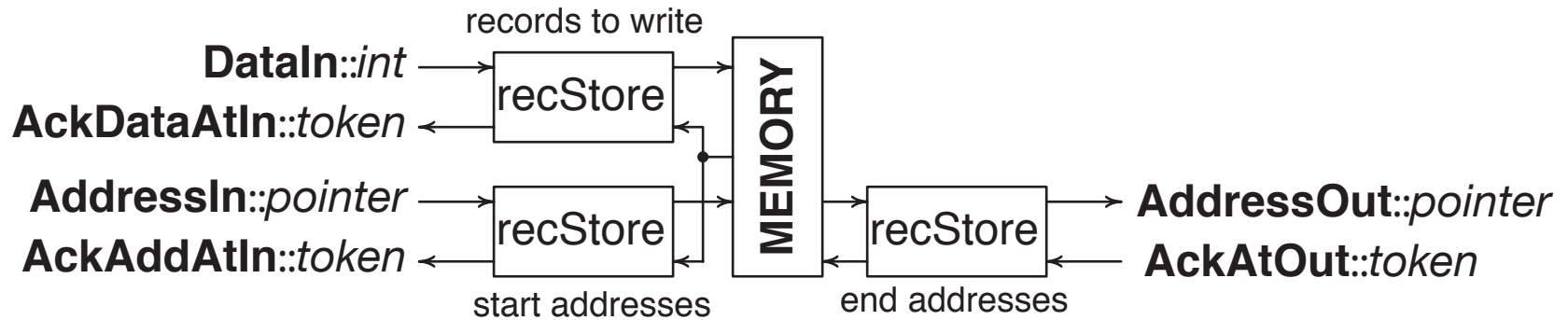
Uses of records



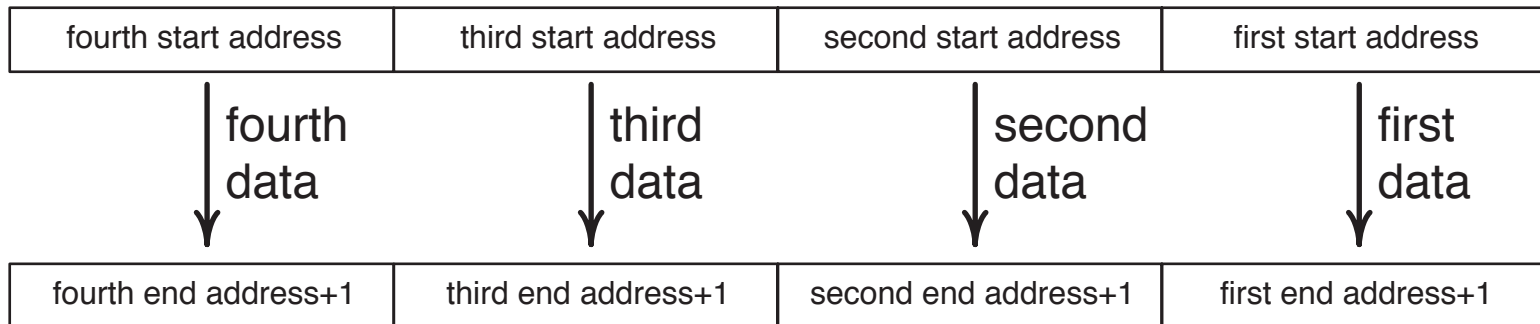
Fetch SHIP



Memory write SHIP

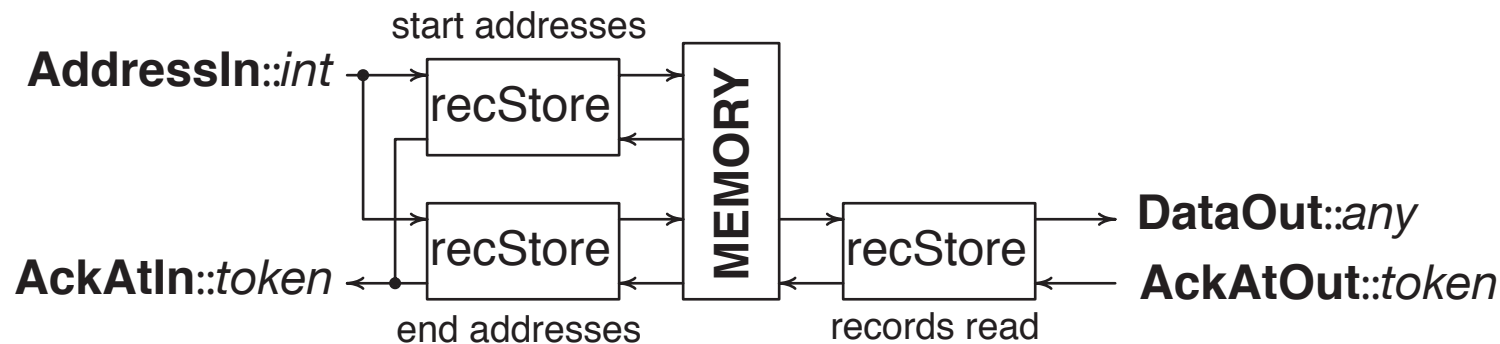


AddressIn

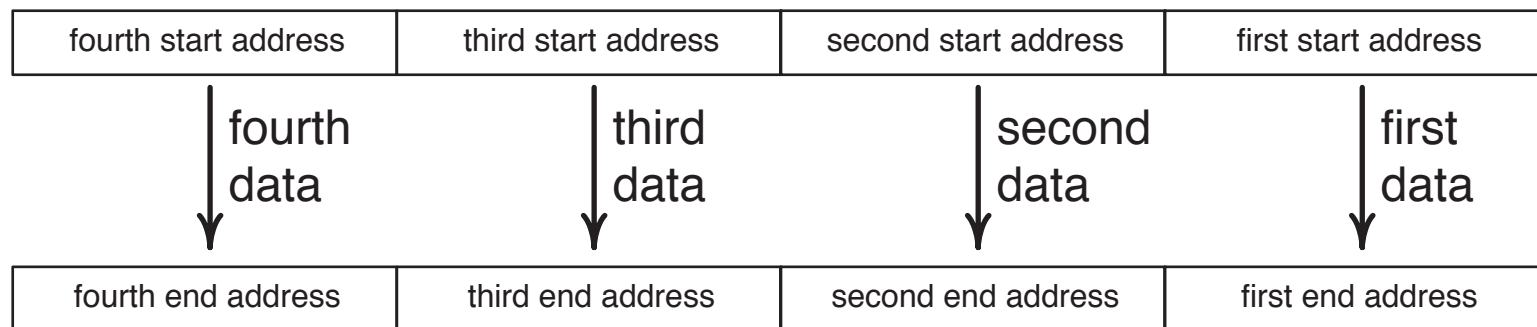


AddressOut

Memory read SHIP

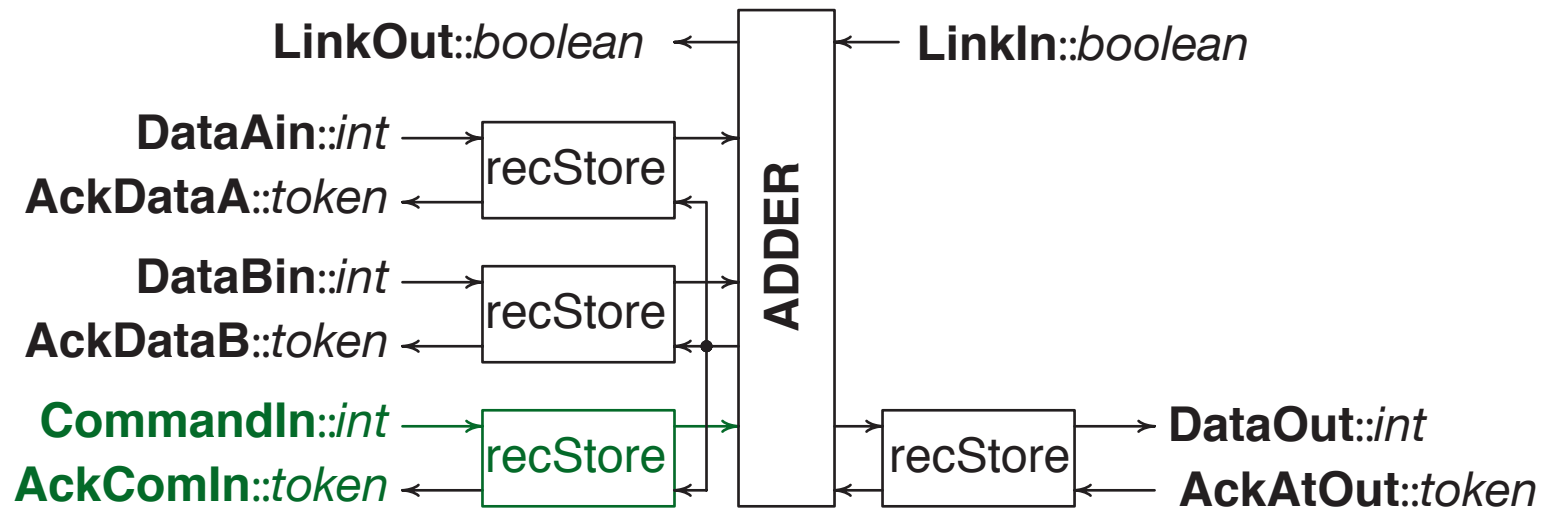


AddressIn



AddressOut

Adder SHIP



LinkOut **Link 3 to 4** **Link 2 to 3** **Link 1 to 2** **LinkIn**



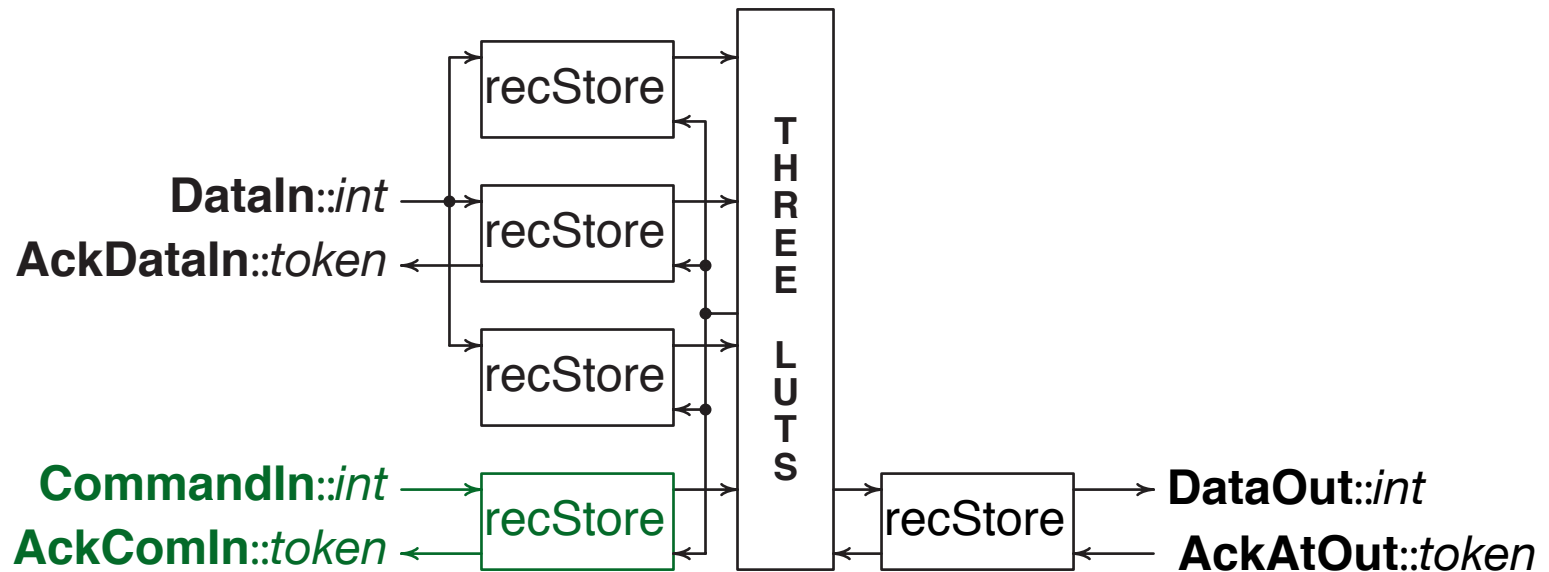
Selection **Link ? A : B** **Link ? A : B** **Link := A > B**

Multiple precision **A+B+Link;**
Link:=carry **A+B+Link;**
Link:=carry **A+B+Link;**
Link:=carry **A+B;**
Link:=carry

Adder SHIP commands

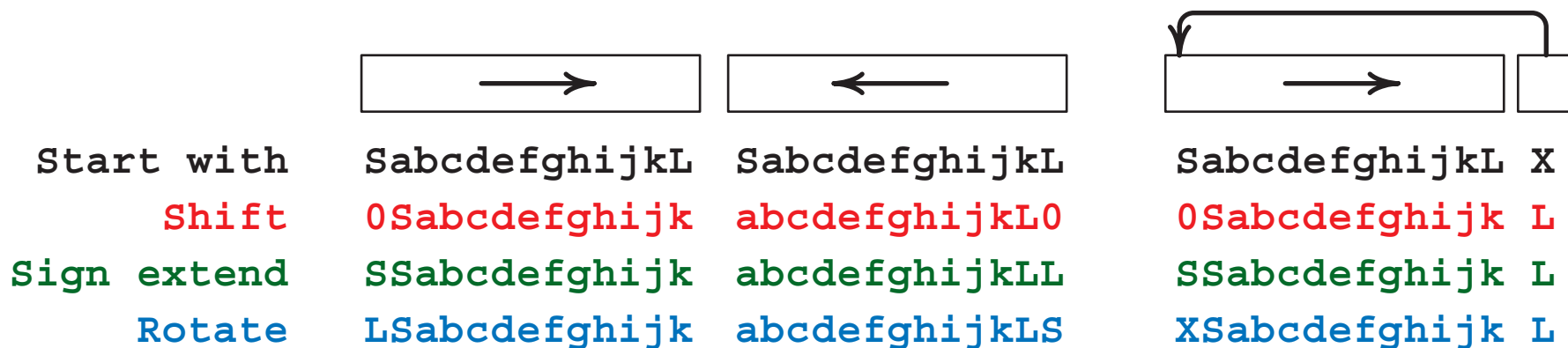
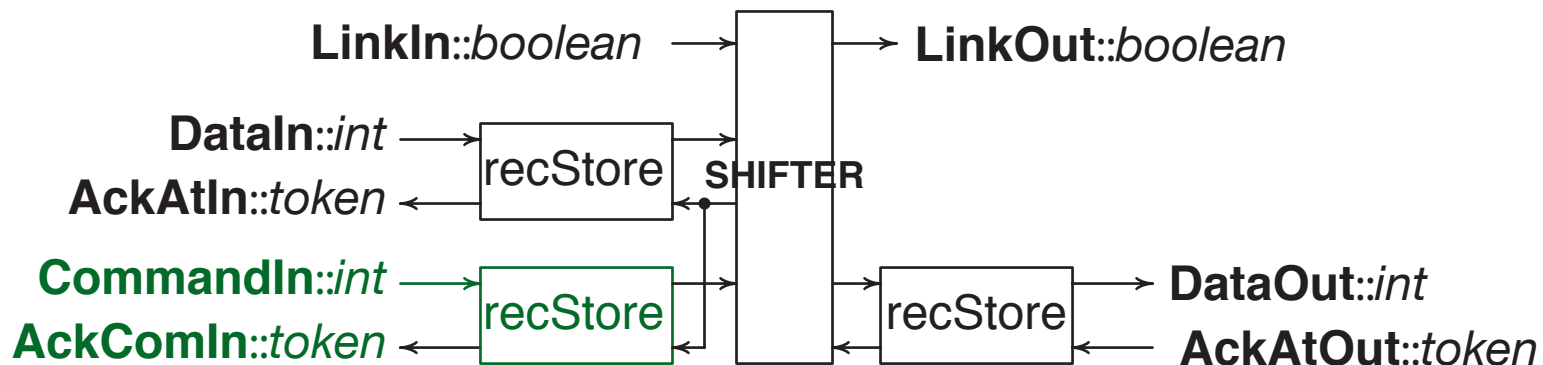
Name	Output	Use LINK (Lin) for	Set LINK (Lout) to
Add	$A + B$	ignore, carry in	Lin, 0, carry, sign, overflow
Subtract	$A - B$	ignore, carry in	Lin, 0, carry, sign, overflow
Copy A	A	ignore	Lin, 0, sign
Copy B	B	ignore	Lin, 0, sign
Select	$Lin?A:B$	select, \neg select	Lin, 0, sign of selected
Negate A	- A	ignore, carry in	Lin, 0, carry, sign, overflow
Negate B	- B	ignore, carry in	Lin, 0, carry, sign, overflow
Magnitude A	A	ignore, carry in	Lin, 0, carry, old sign
Magnitude B	B	ignore, carry in	Lin, 0, carry, old sign
Larger	$\max(A,B)$	ignore, carry in	Lin, 0, carry, $A > B$
Smaller	$\min(A,B)$	ignore, carry in	Lin, 0, carry, $A < B$
Increment A	$A + 1$	ignore, carry in	Lin, 0, carry, sign, overflow
Increment B	$B + 1$	ignore, carry in	Lin, 0, carry, sign, overflow

Bitwise SHIP



fourth value A	third value A	second value A	first value A
fourth value B	third value B	second value B	first value B
fourth value C	third value C	second value C	first value C
fourth command	third command	second command	first command

Shift & rotate SHIP



Shift and Rotate

Name	Output	Use LINK (Lin) for	Set LINK (Lout) to
Copy	A	ignore	Lin, 0, 1, sign
Shift left	$A \ll 1$	ignore, carry, extend	Lin, 0, 1, old sign
Shift right	$A \gg 1$	ignore, sign in, extend	Lin, 0, 1, old LSB
Rotate left	circular	ignore	Lin, 0, 1, old sign
Rotate right	circular	ignore	Lin, 0, 1, old LSB
Rotate left w/link	as N+1 bits	rotate in	Lin, 0, 1, old sign
Rotate right w/link	as N+1 bits	rotate in	Lin, 0, 1, old LSB
Select left shift	Lin? $A \ll 1 : A$	shift control	Lin, 0, 1, old sign
Select right shift	Lin? $A \gg 1 : A$	shift control	Lin, 0, 1, old LSB