

# **FLEETzero:**

## **An Asynchronous Switching Experiment**

William Coates  
Jon Lexau  
Ian Jones  
Scott Fairbanks  
Ivan Sutherland

**Sun Microsystems Laboratories**  
**Mountain View, CA**

**ASYNC-2001**  
**Salt Lake City**  
**March 12, 2001**

# SUMMARIZE

- 1) *GasP: A Minimal FIFO Control*
- 2) *Fast Asynchronous Circuits*
- 3) *Squaring the FIFO in GasP*

# FLEETzero

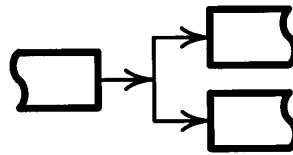
# IMPLICATIONS

# GasP: MODULES FOR:

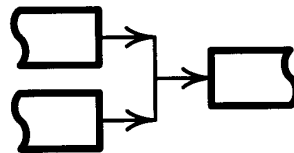
**Pipeline**



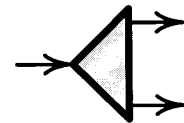
**Unconditional  
branch**



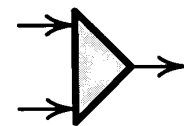
**Unconditional  
merge**



**Data-conditional branch**

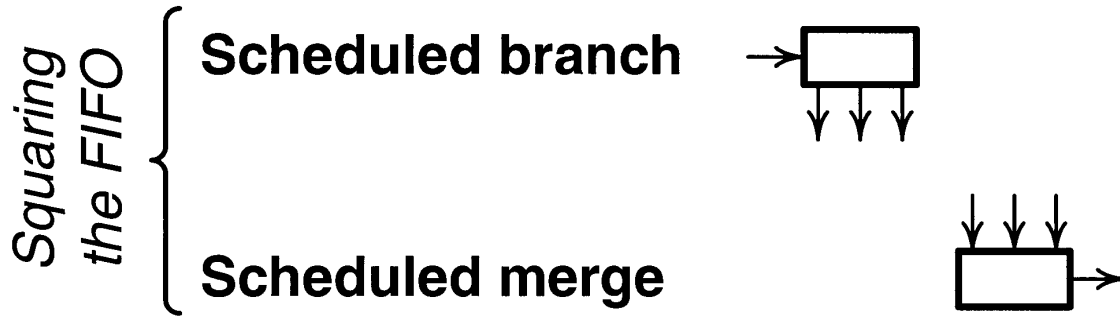


**Data-conditional merge**



→ = n data + 1 control wire

# MORE GasP MODULES:



→ = n data + 1 control wire

# DESIGN SEQUENCE

- 1) Promise uniform stage delay
- 2) Make loops uniform length
- 3) Lay out and measure wires
- 4) Pick transistor widths  
for uniform stage delay
- 5) Confirm with simulation

## RESULT

**FAST - FAST - FAST**

**Like 3-inverter ring oscillator**

**1.5 GDI/s in 0.35 micron**



# **UNIFORM STAGE DELAY**

**By careful choice of transistor widths**

**So each transistor drives a load  
proportional to its width**

**Requires accurate wire loads**

**Encourages self-resetting designs**

**Robust over voltage variation  
(and temperature?)**

**Set actual delay after layout**

**Speed vs power tradeoff**

# OBSERVATION

**"Modern" Computers are  
too much like those of the 50s**

**Logic WAS expensive  
Wires WERE cheap**

**Logic IS cheap  
Wires ARE expensive**

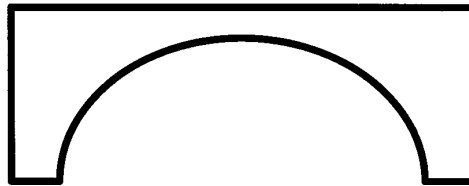
**Legacy of obsolete ideas**

**Much effort in design to retain  
"compatibility"**

**Compatability not real anyway**

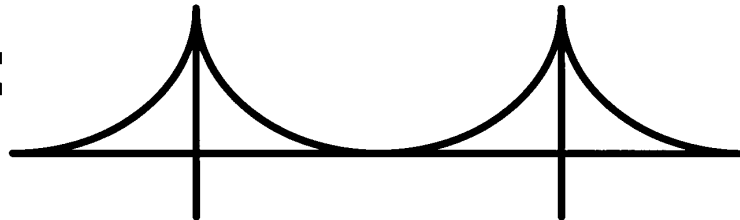
# OUR THRUST

**DON'T:**



**"asynchronize" existing structures**

**DO:**



**use asynchrony for new structures**



# **PRINCIPLE:**

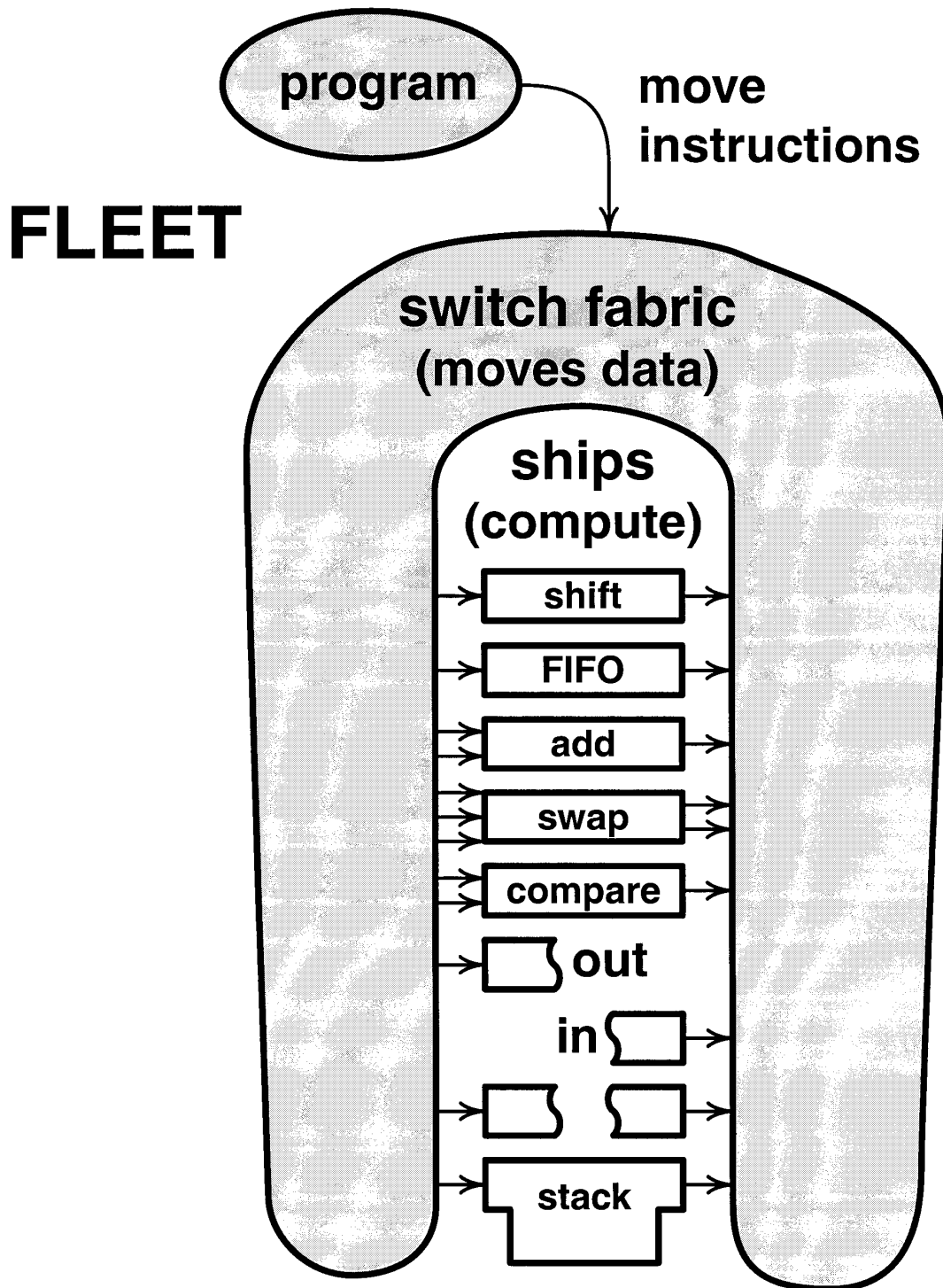
**Moving data is  
the costly part.**

**∴**

**Put the programmer  
in charge of data moves.**

## ***MOVE A to B***

**What happens depends on B.**



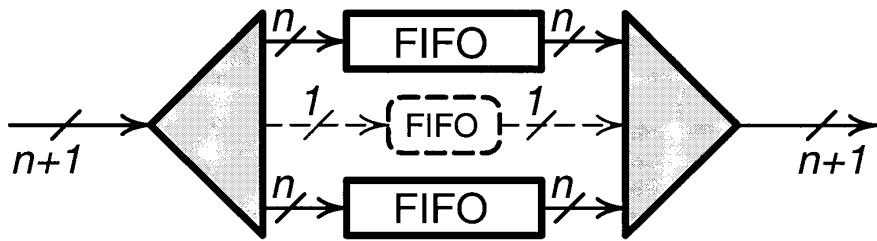
# **PROBLEM:**

**How to select 1 out of N  
asynchronously.**

**Was too hard with transistors.**

# AH HA! = Eureka

----> = "order wire"

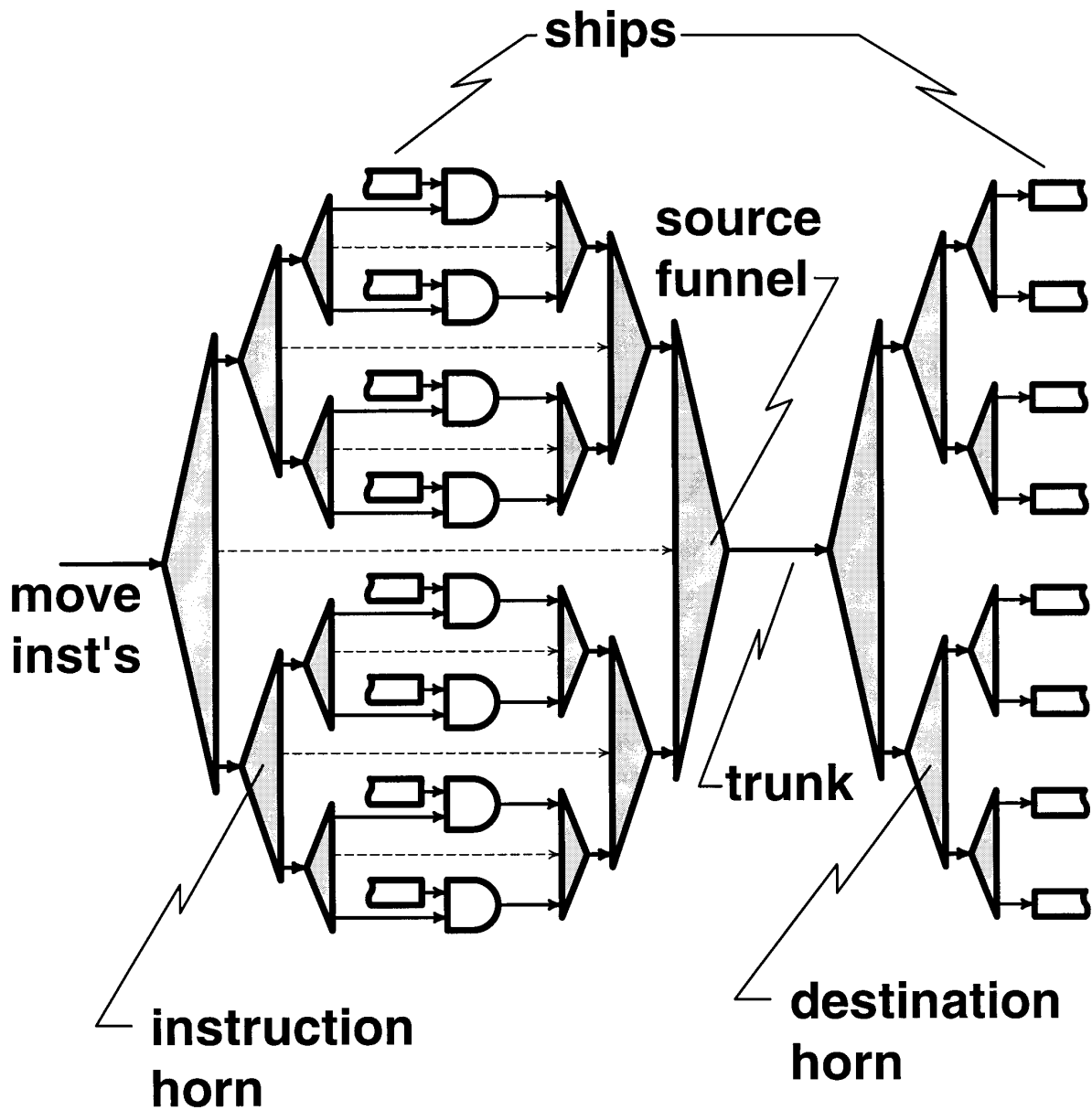


is same as

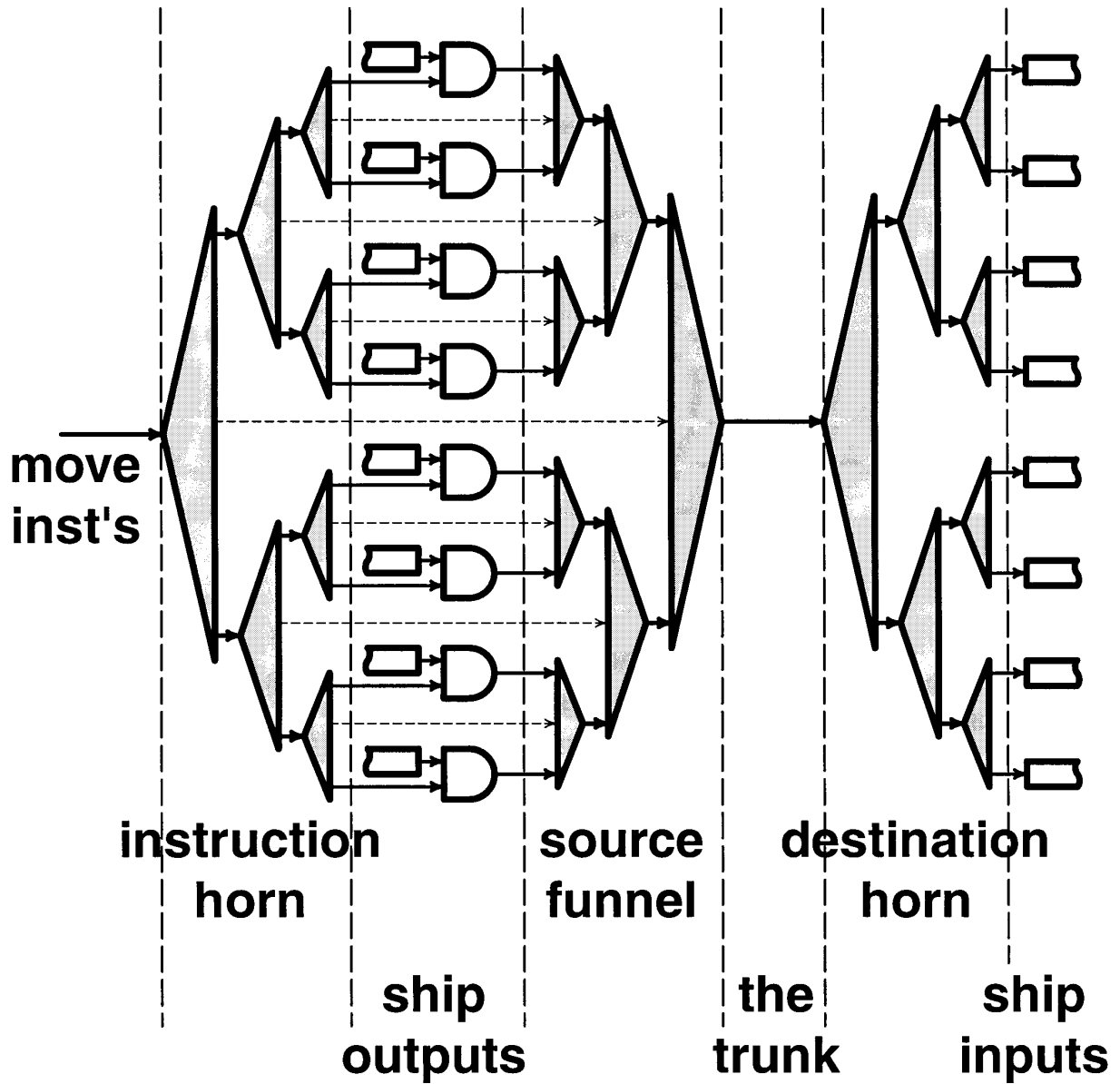


this definition is recursive

# FLEETzero SWITCH FABRIC



# FLEETzero SWITCH FABRIC



# **FLEETzero:**

**Tests the switch fabric**

**Up to  $1.5 * 10^9$  instructions/sec**

**Uses double ring source**

**no program branch**

**min < length of loop < max**

**Has a variety of "ships"**

**FIFOs**

**Adder**

**I/O**

# **FLEETzero is a TOY,**

**8 bit words**

**8 ships**

**No program branch**

**No memory**

## **but offers a new paradigm:**

**Data - centric view of computing**

**Highly concurrent**

**Self - timed**

**FAST - FAST - FAST**



# **NEW PARADIGM**

**Listen to silicon.**

**Wires are expensive.**

**Asynchronous flexibility.**

**Incremental upgrade.**

**Timing to suit the function.**

# **SUMMARY**

**GasP - simple circuit modules**

**Sizing - uniform stage delays**

**Square FIFO - state machines**

**FLEET - computing paradigm**

**"GasP is just a clock  
generation and  
distribution scheme."**