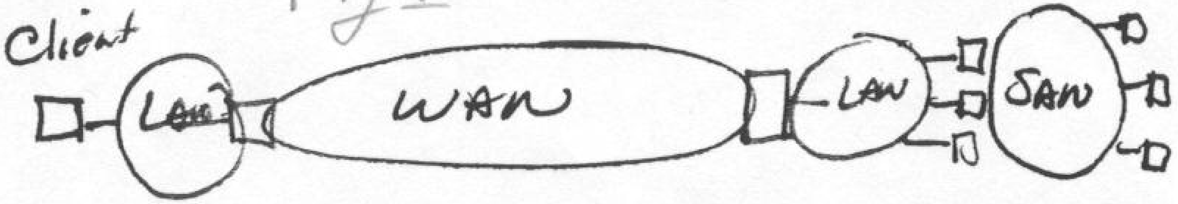
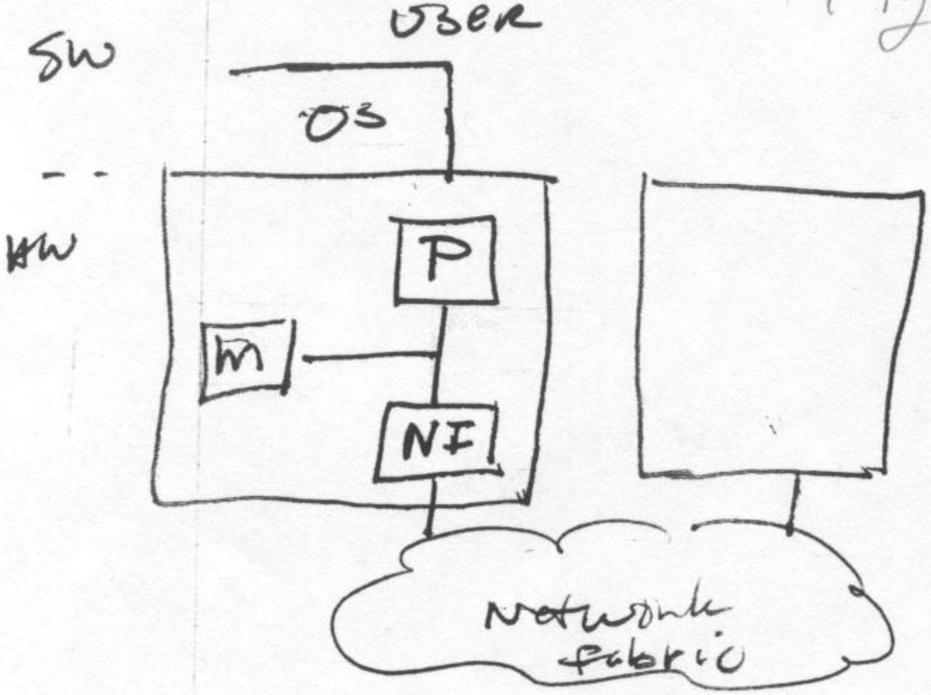
Fig I



What's a network system?

Fig 2

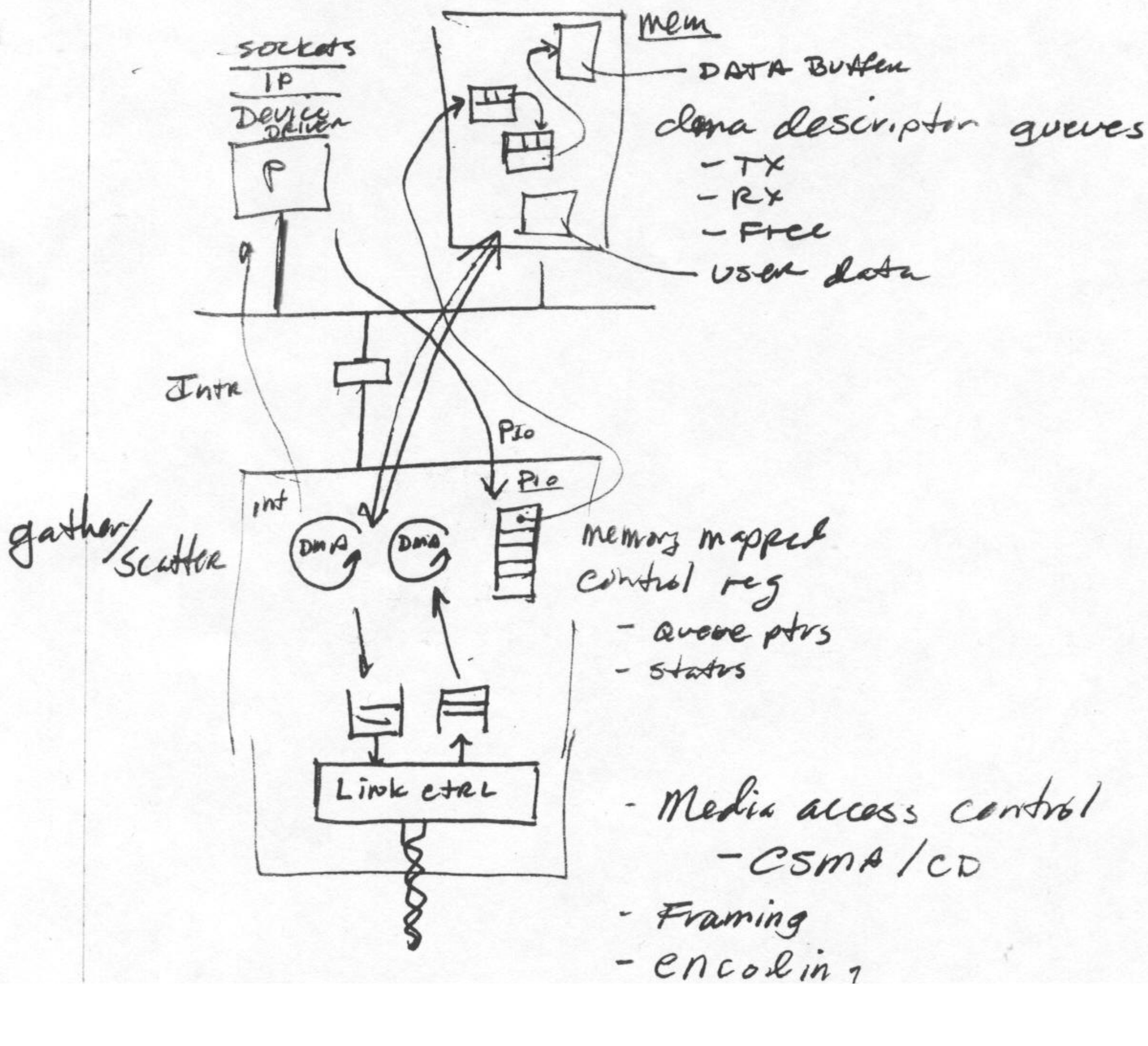


FB3

Typical LAN NIC

Issues:

- juggling events
- accessing data & control info
- copy (blind data receive)



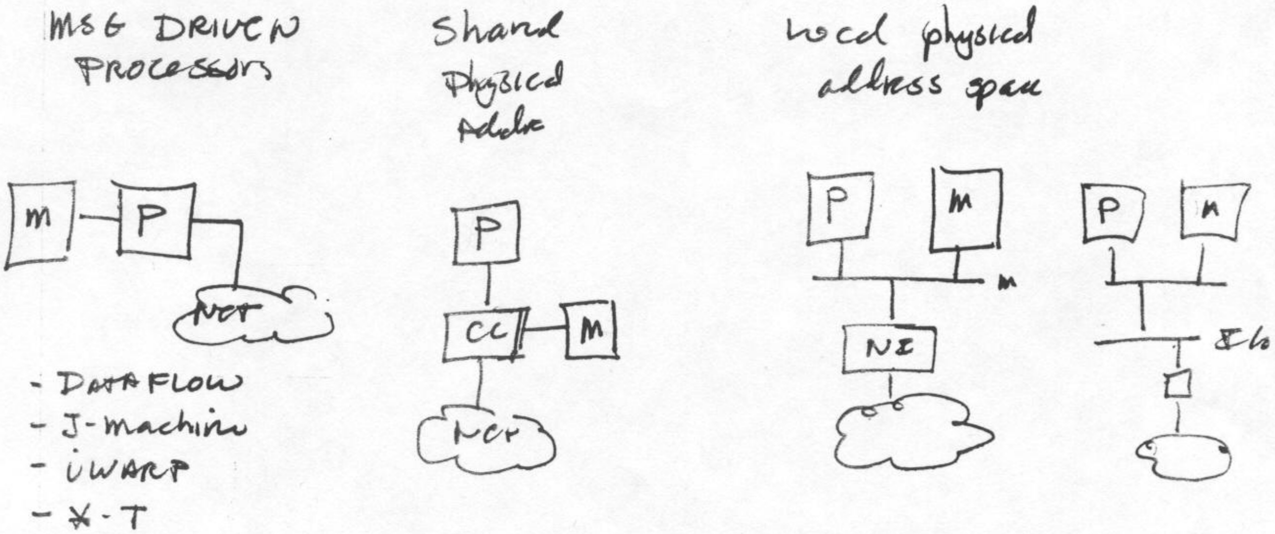
dma descriptor queues
- TX
- RX
- Free

memory mapped control reg
- queue ptrs
- status

- Media access control
  - CSMA/CD
- Framing
- encoding

Parallel Machine Networks

Fig 4



SIMPLE DMA

Fig 5

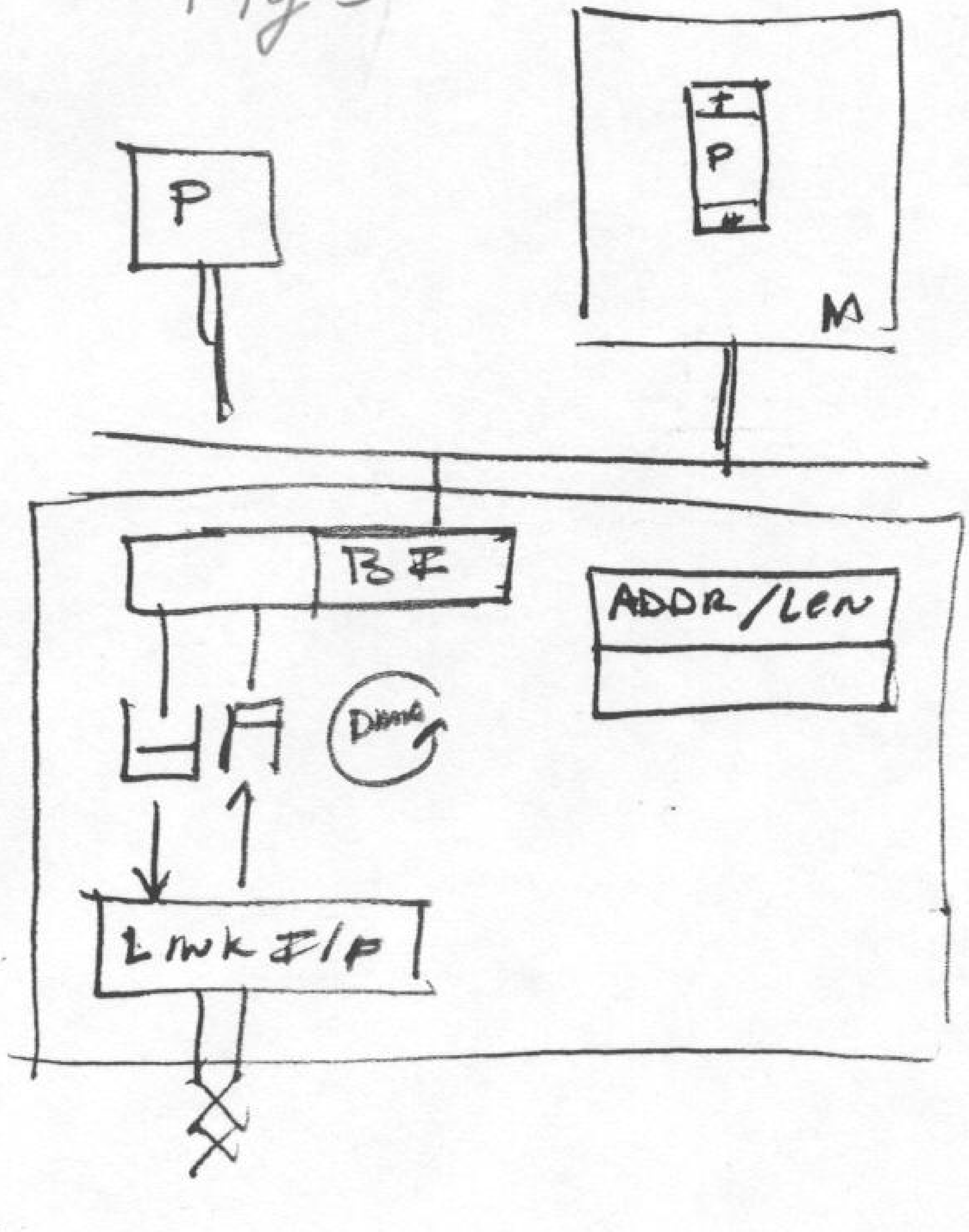


Fig 6

SIMPLE PIO

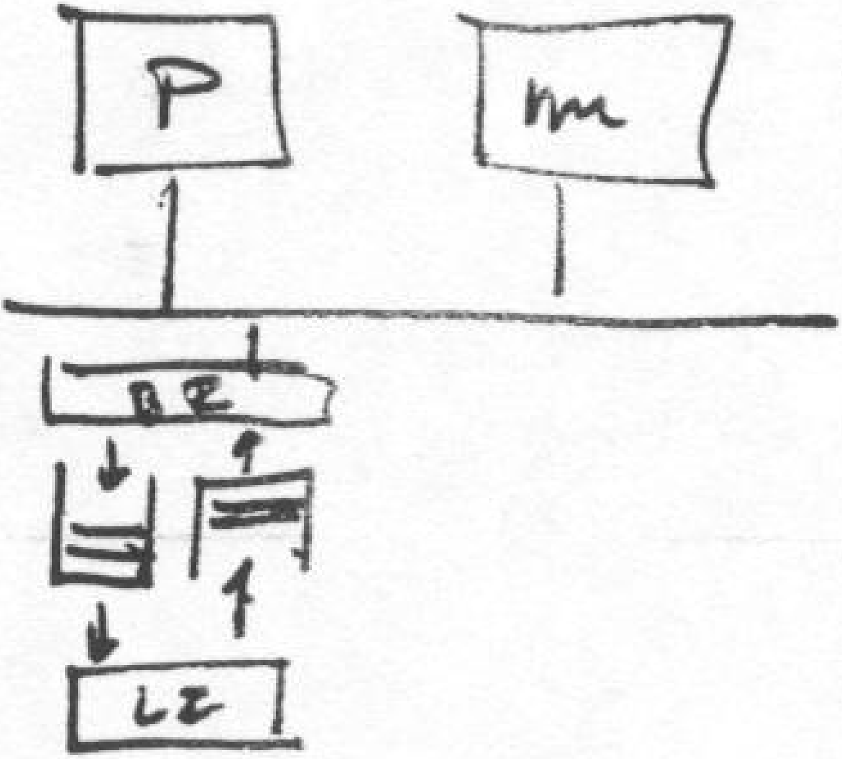


Fig 7

Physical Parallelism

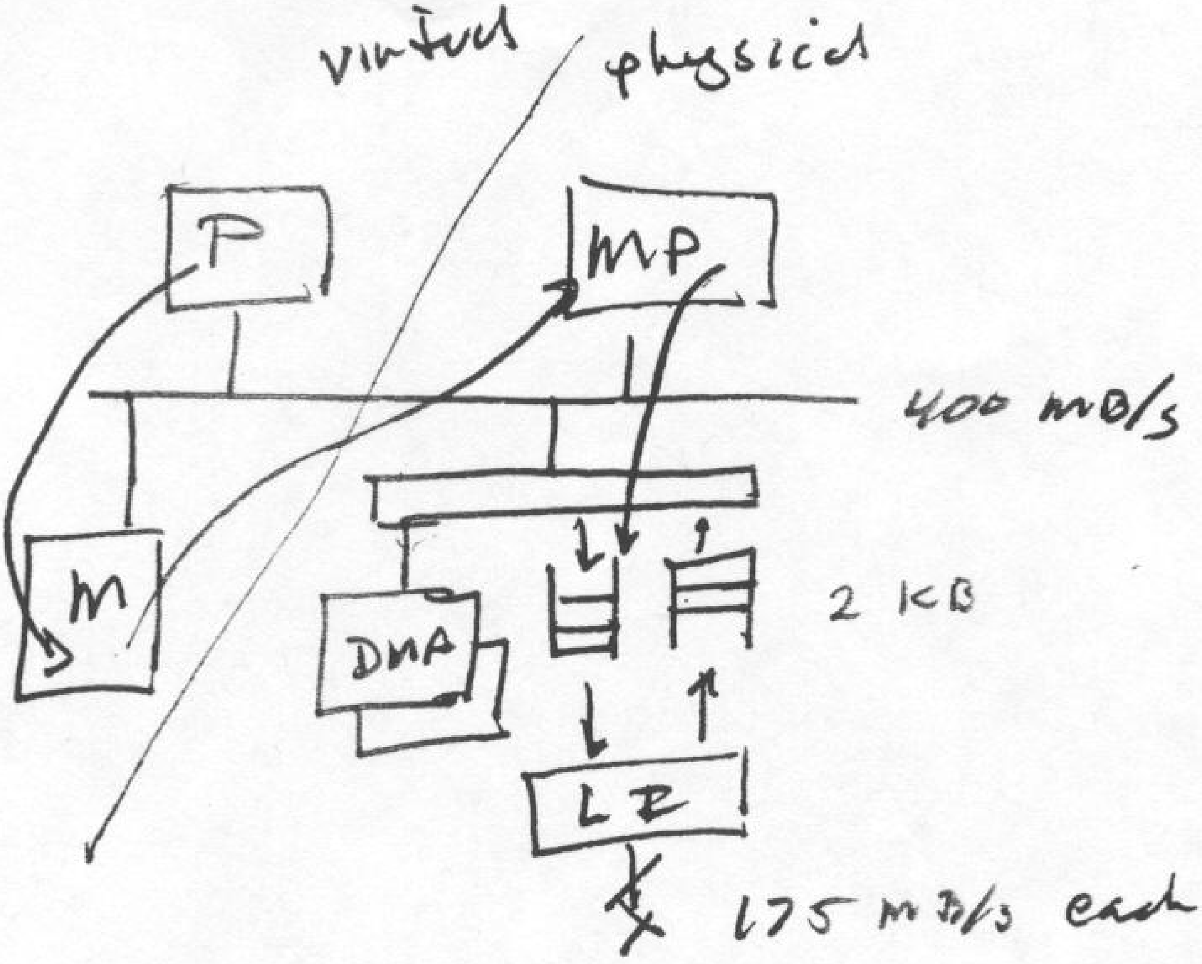


Fig 8

Virtual (Multithreaded) Parallelism

Ex: Meiko

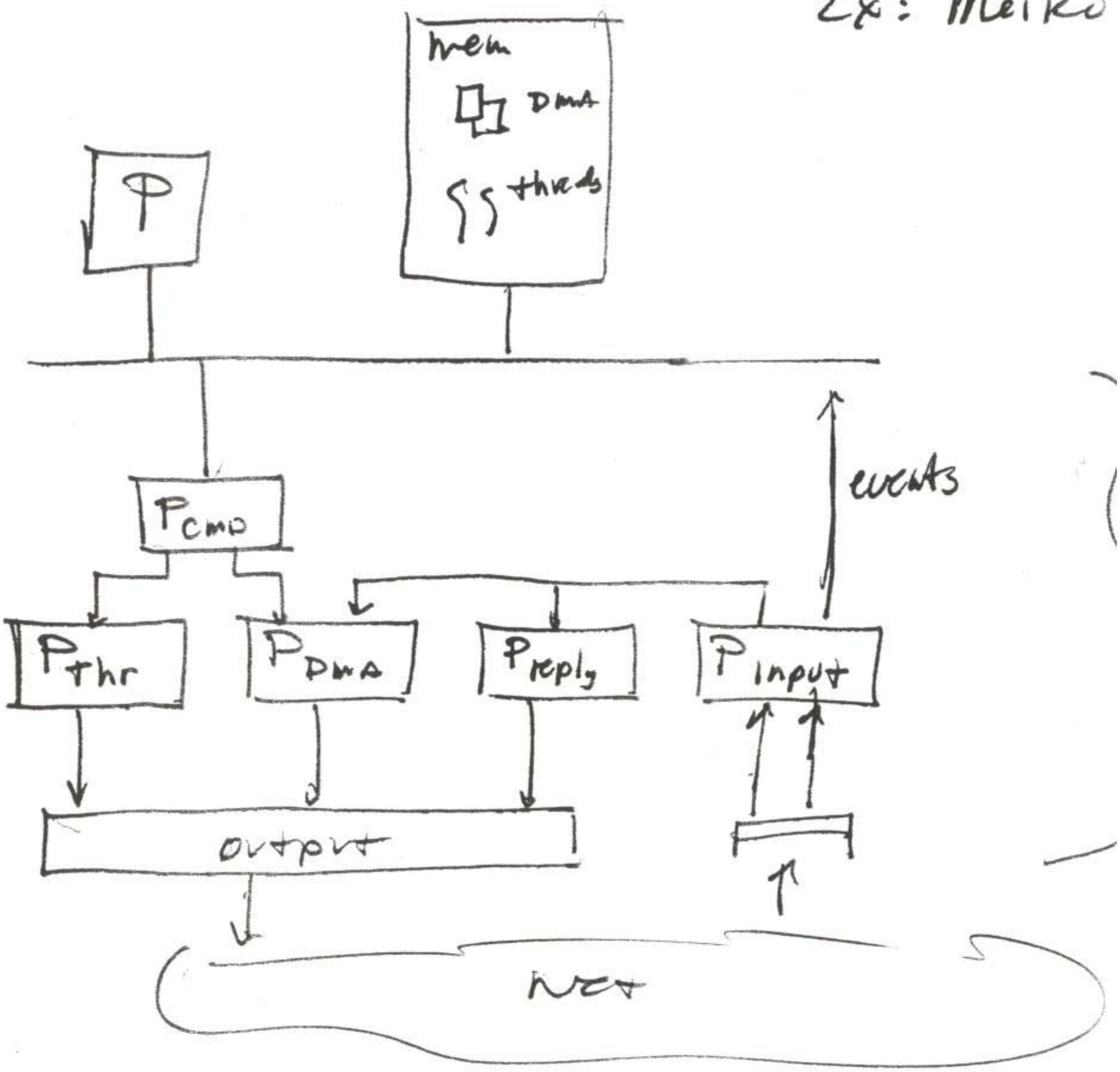
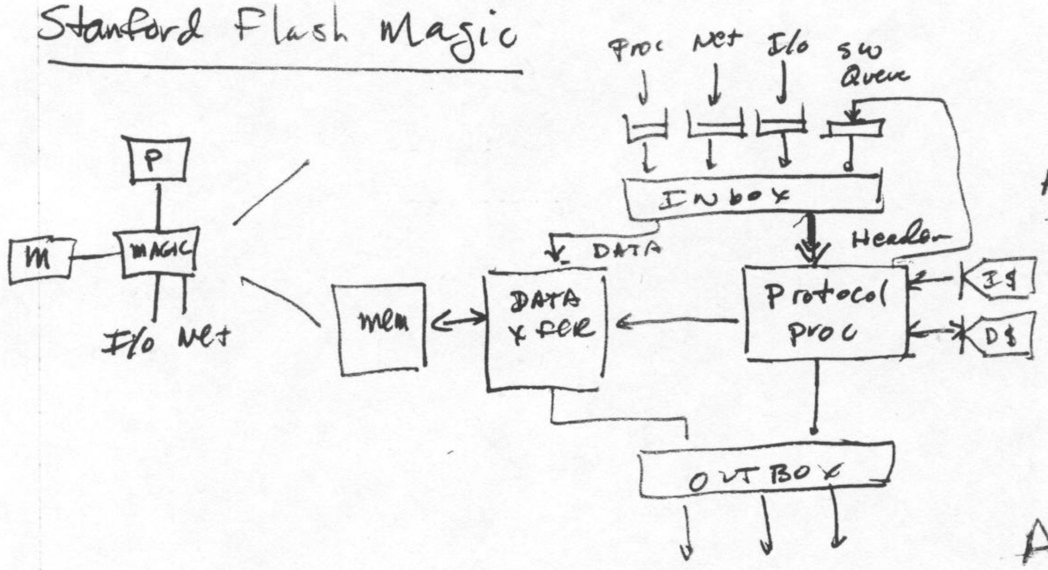


Fig 9

Stanford Flash Magic

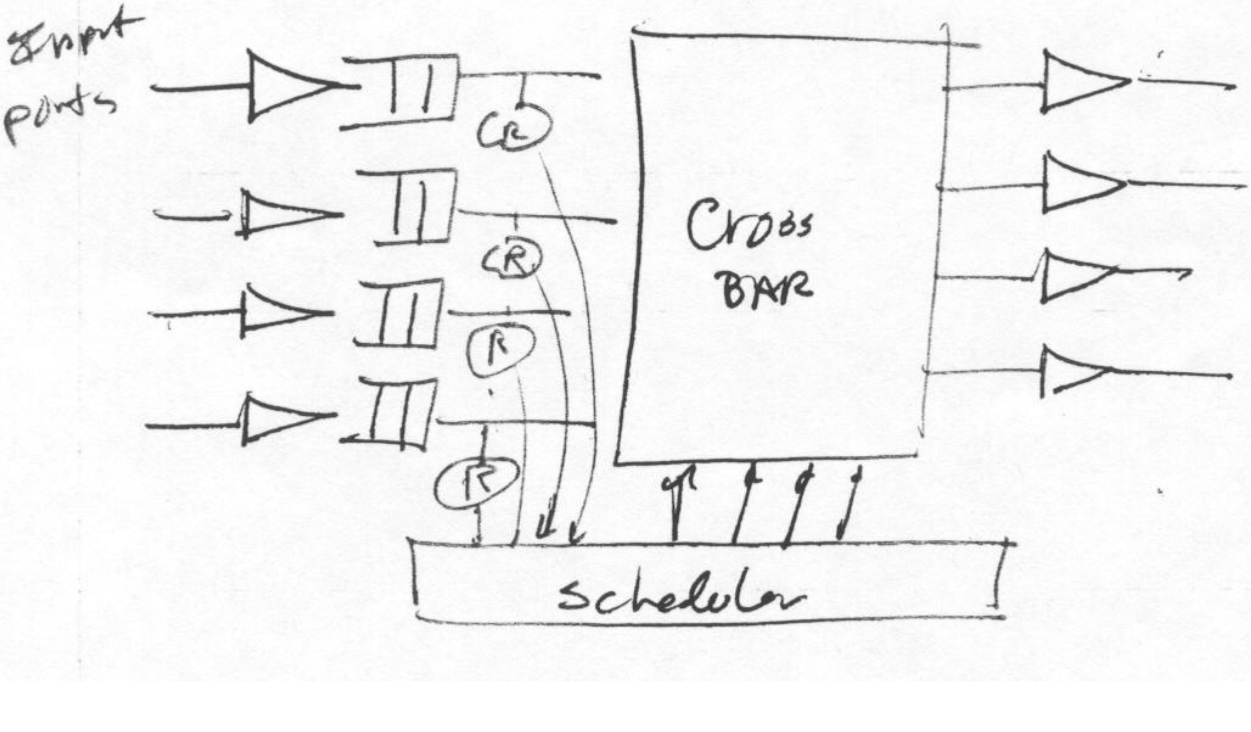


Handler dispatch

Also in *T

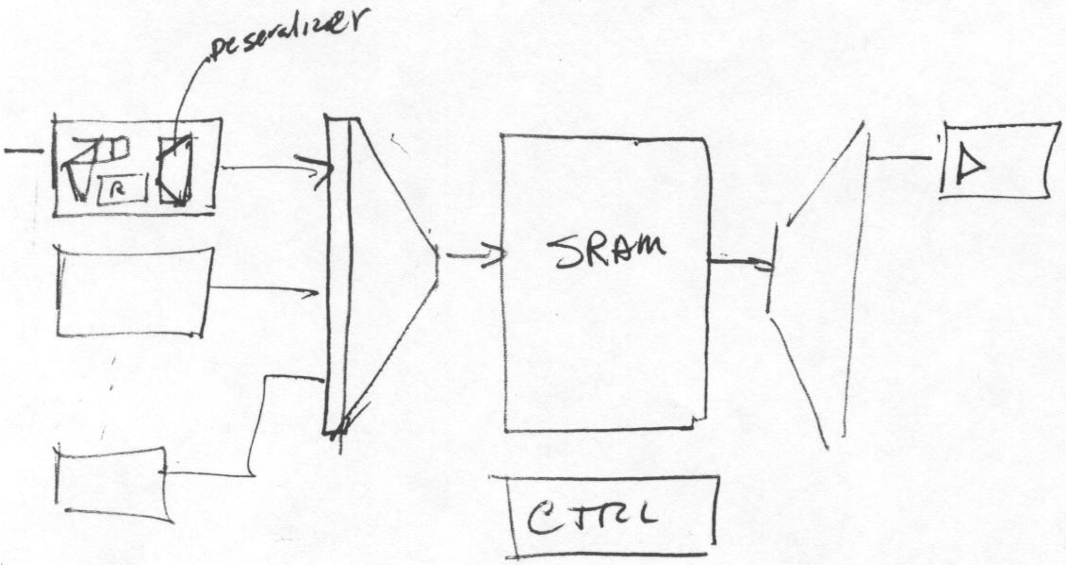
Classic View

Fig10



'Cache-like' switch design

Fig 11



1 Gb/s links ⇒ 128 bit chunks @ 8 mHz per port

8 ports ⇒ 64 mHz

Scans of original lecture notes (Culler)

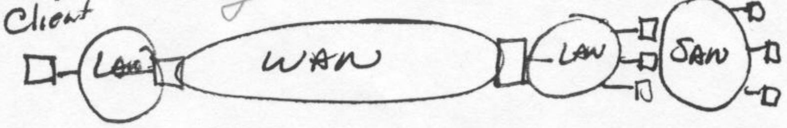
Network Interface Architecture CS252 2/21/

Motivation

- networks becoming the cannonical I/o device
- Blow the machine apart
- networked appliances

- Internet architectures are all about networks

Fig I



- Parallel machine Networks ⇒ LAN / WAN Breakthrough
  - design approach
  - "killer switch"

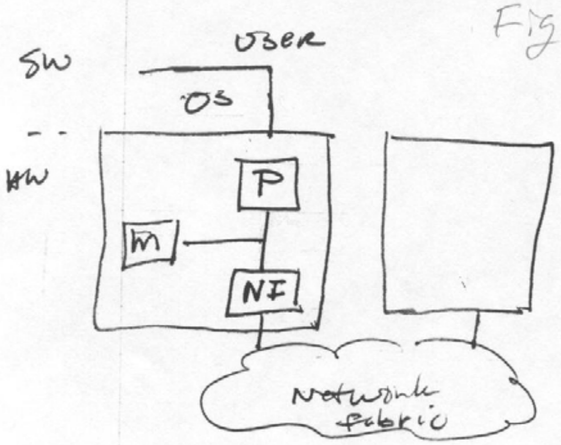
- Different Kind of Architecture Challenge
  - juggling events & data movement

Today:

LANS vs PARALLEL MACHINES vs WIRELESS

What's a network system?

Fig 2



Issues

- How is communication integrated into node arch?
    - processing & storage
- What is the communication abstraction?
    - o/s calls, library, PIO, load/store
- What is the NI architecture?
- How does data get from place to place?

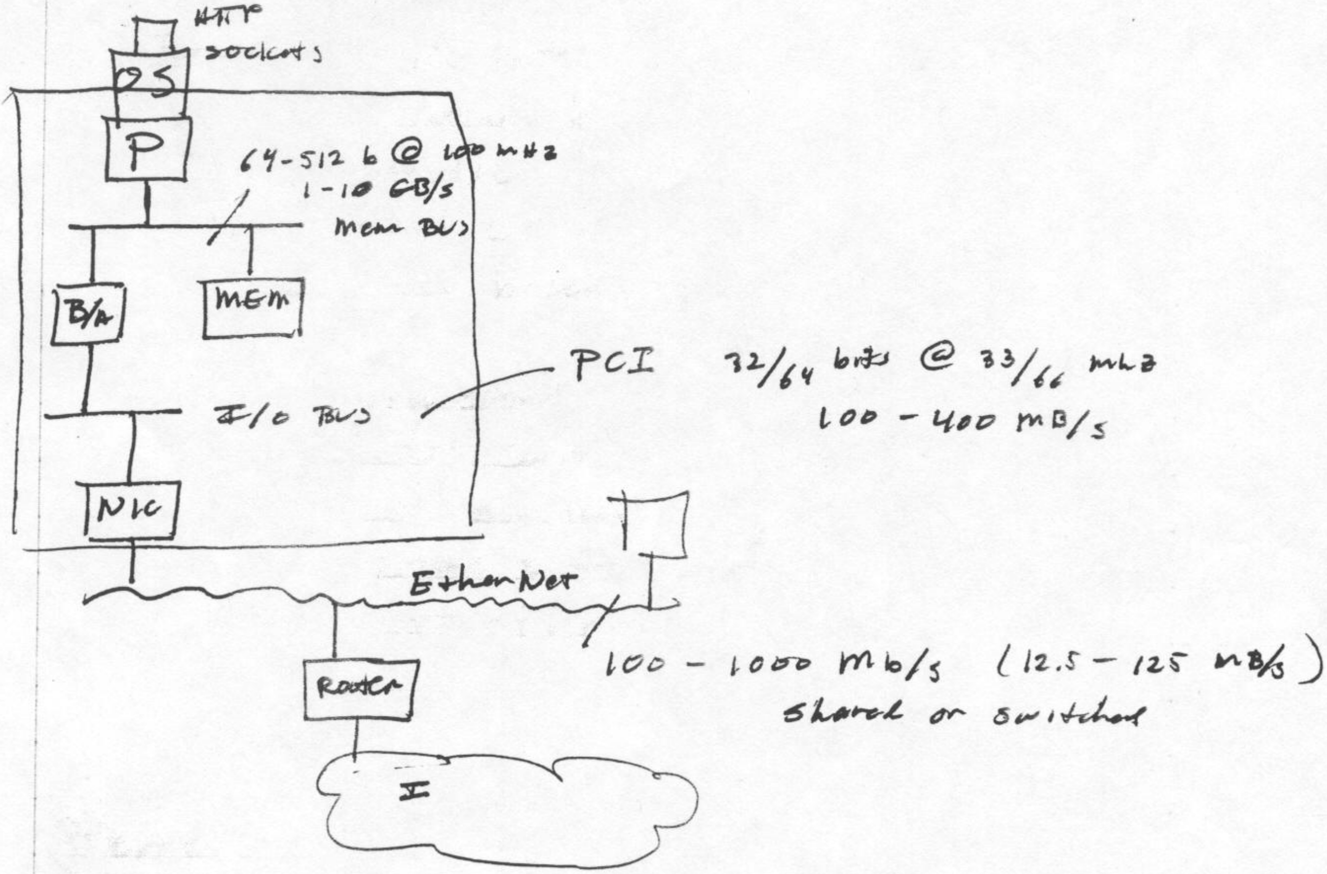
PARALLEL MAchines SANS LANS WANS

Performance framework

$$\text{Send overhead} + \text{time of Flight} + \underbrace{\frac{\text{Size}}{\text{BW}}}_{\text{Transfer time}} + \text{RCV overhead}$$

$$\text{Eff BW} = \text{BW} \cdot \frac{\text{Xfer Time}}{\text{Total Time}}$$

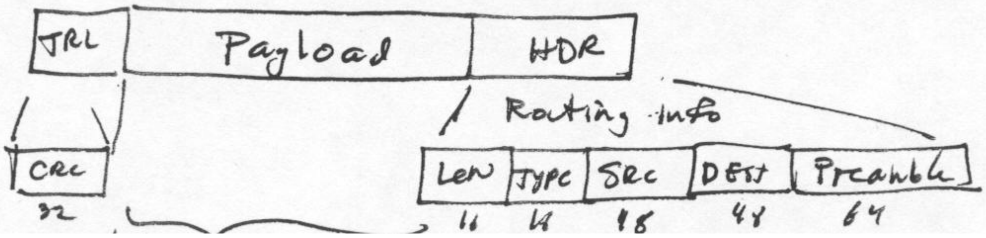
Classical LAN



Key Abstractions

- digital symbols :
  - channel coding (manchester)

- Framing - information is serialized



Kinds of routing

Determing the route
- destination based
- arithmetic
- virtual circuit
- source based

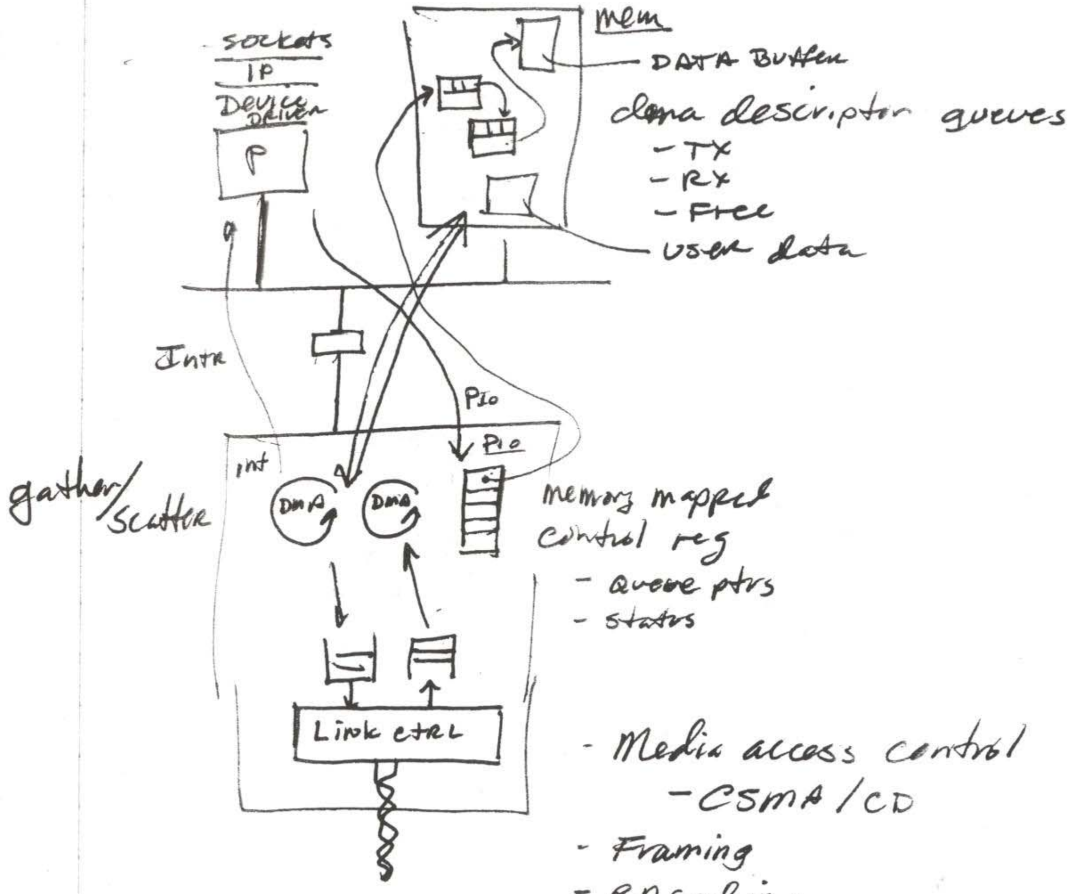
Mode of transport
- circuit switched
- packet switched
  - store & forward
  - cut-through

Performance

FB3

Typical LAN NIC

Issues:
- juggling events
- accessing data & control info
- copy (blind data receive)



- Media access control
  - CSMA/CD
- Framing
- encoding

} completely standardized

Typical costs

Total time
Overhead
Transfer time

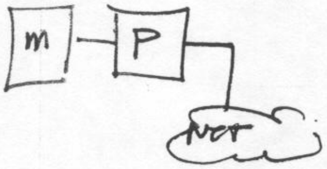
} ⟹ Reduce Overhead
- zero copy

User level Interface
- protection

PARALLEL MACHINE Networks

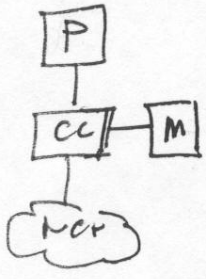
Fig 4

MSG DRIVEN PROCESSORS

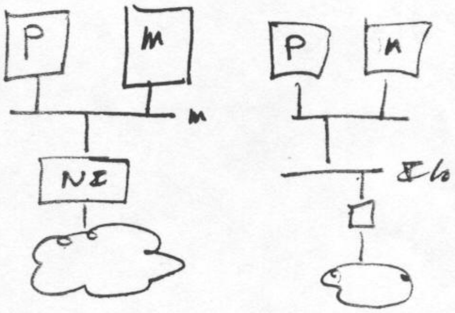


- DATAFLOW
- J-machine
- iWARP
- *-T

Shared Physical Addr



Local physical address space

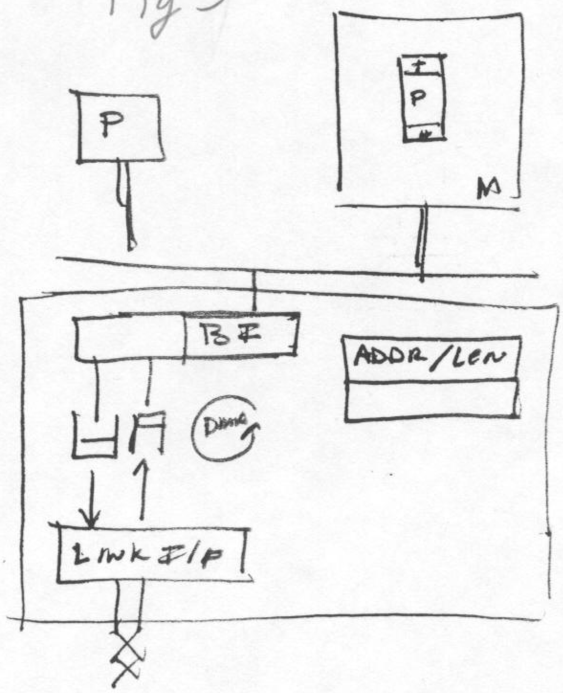


Very low latency network
High BW
Regular topology
Highly reliable

} ⇒ minimize overhead

SIMPLE DMA

Fig 5

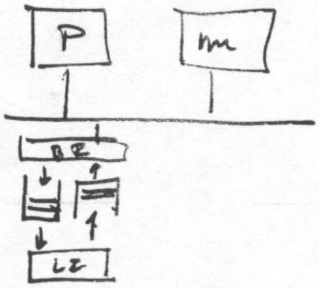


ex: nCube, ipsc/2

- Simple Framing
- ~~[illegible]~~
- few stores to start xfer
- Copy or swap receive data
- virtual ↔ physical

Fig 6

SIMPLE PIO



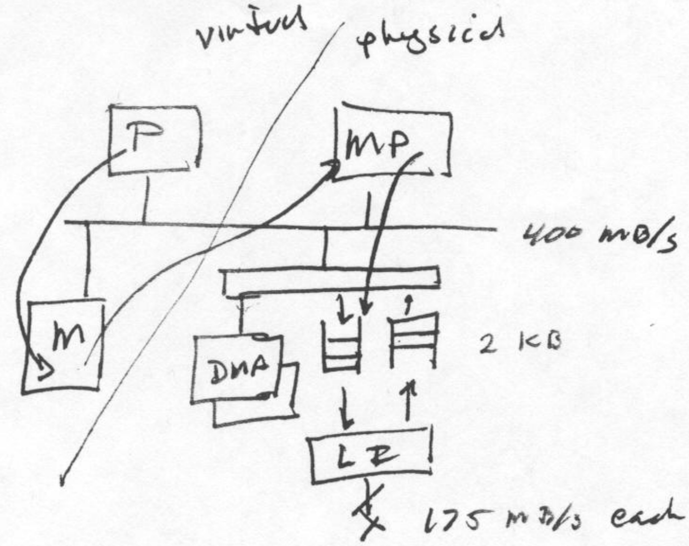
ex: CM-5

- No copy
- Fast small msgs
- limited BW
- Proc. intensive
- OS manages NI
  - process mapping
  - context switch

- Host code dispatches on many conditions

Fig 7

Physical Parallelism



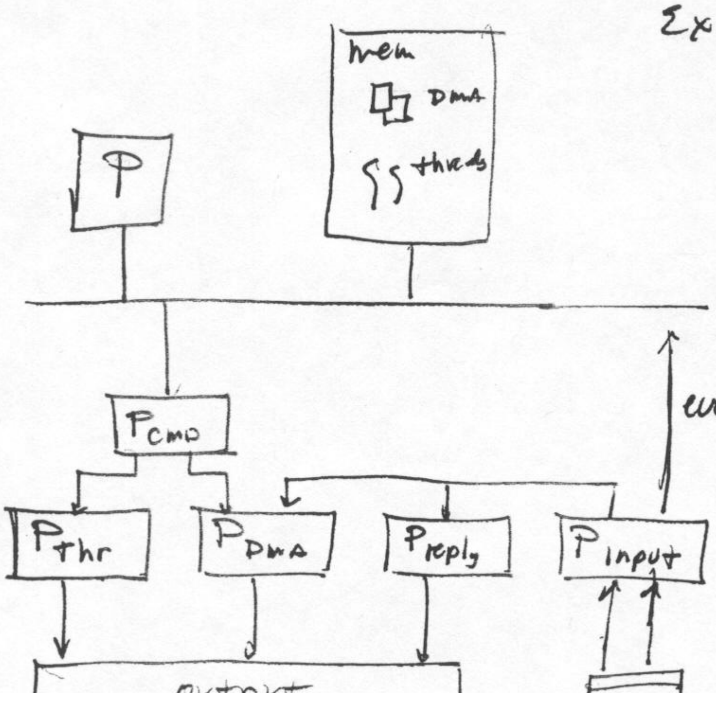
Ex: Intel Paragon

- concurrency intensive
  - proc events
  - NI events
  - DMA events
- BW intensive
  2-3 x bus crossing
- synch overhead

Fig 8

Virtual (Multithreaded) Parallelism

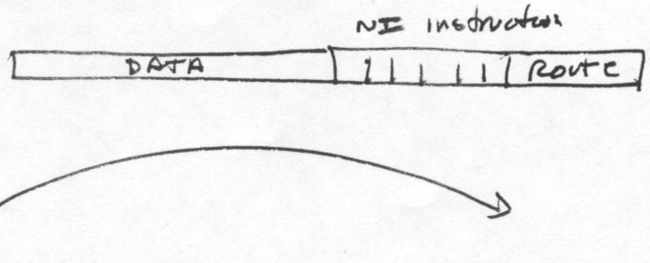
Ex: Meiko CS-2



Implemented on single μ-code engine

- zero data transfer cost

Network Transaction

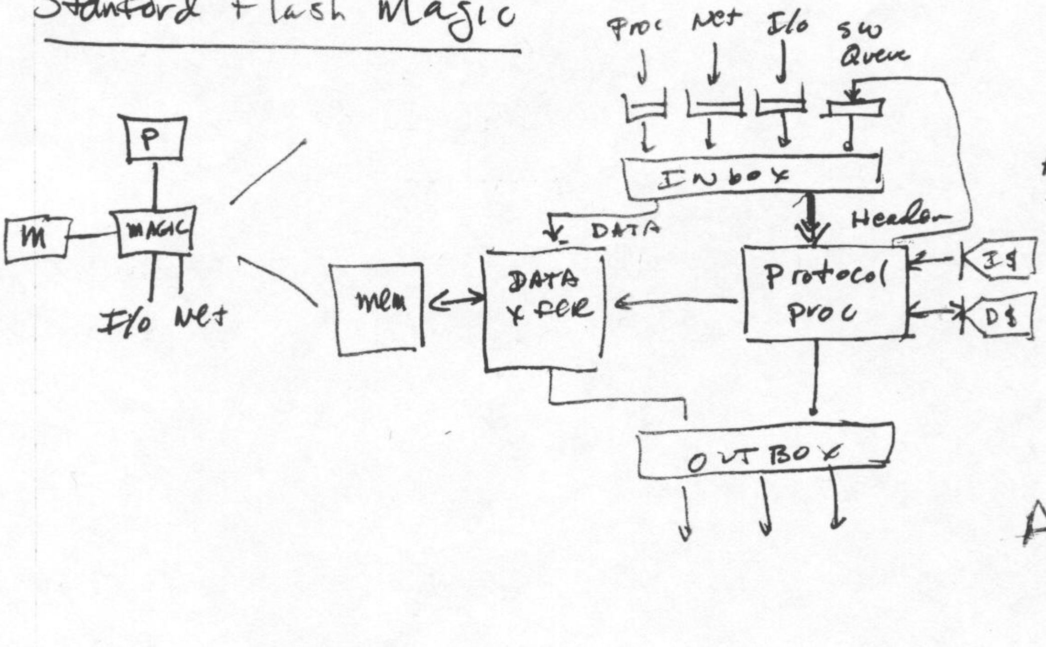


DMA TX - builds "store data" cmd
Thr - straight-line code
- abort on error (incl page fault)
Remote Rd - DMA reply

Flexible protocols

Fig 9

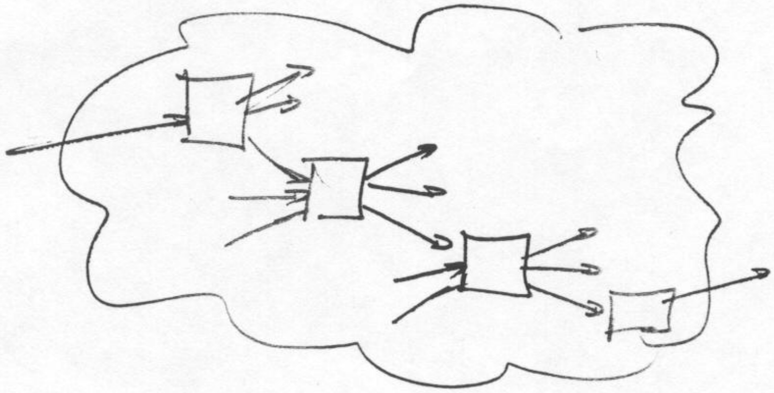
Stanford Flash Magic



Handler dispatch

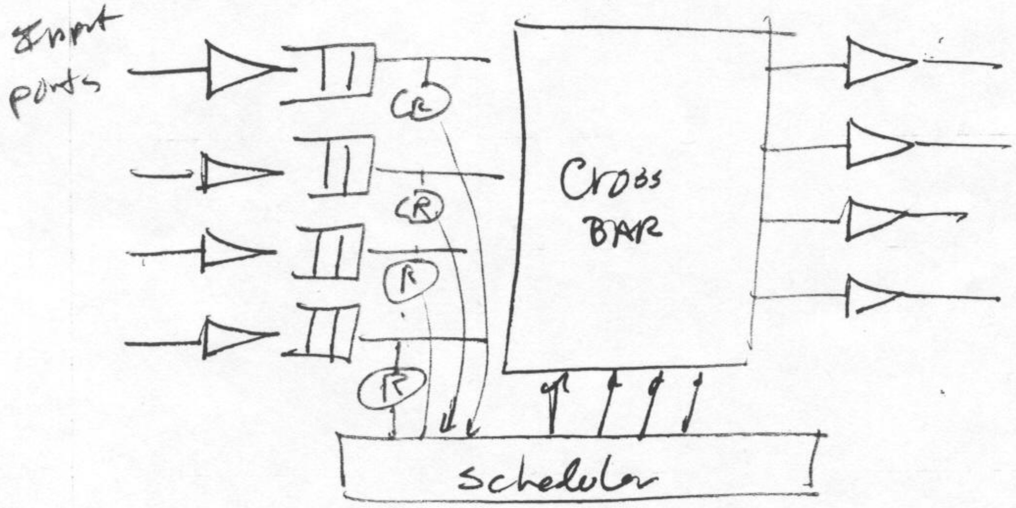
Also in *J

Routing / Switching



Classic View

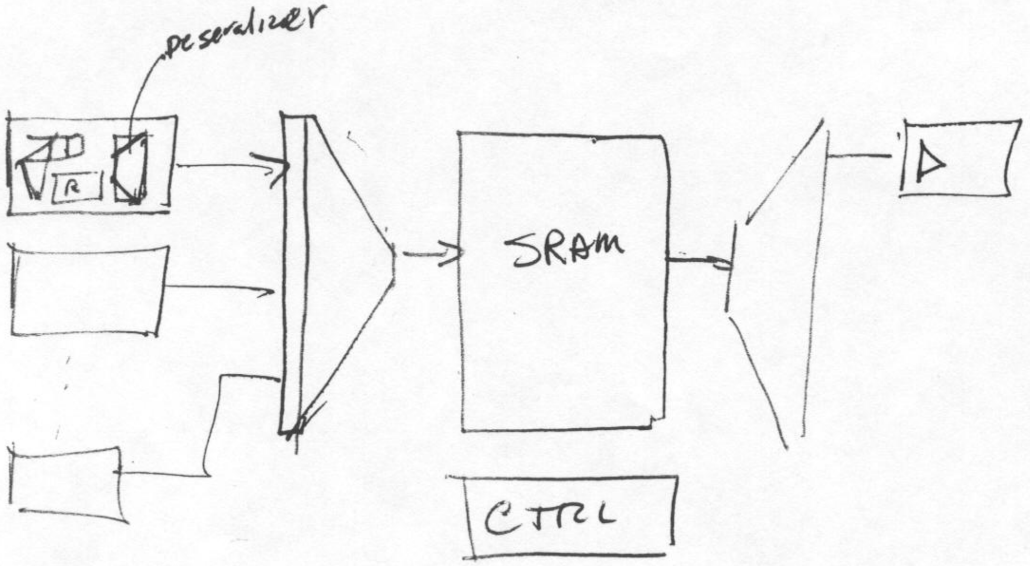
Fig 10



- Recognize Packet
- Compute output port
  - Arithmetic in mesh, butterfly, etc
  - select in source based
  - table look up in dest or virtual circuit

'Cache-like' switch design

Fig 11



1 Gb/s links ⇒ 128 bit chunks @ 8 mHz per port

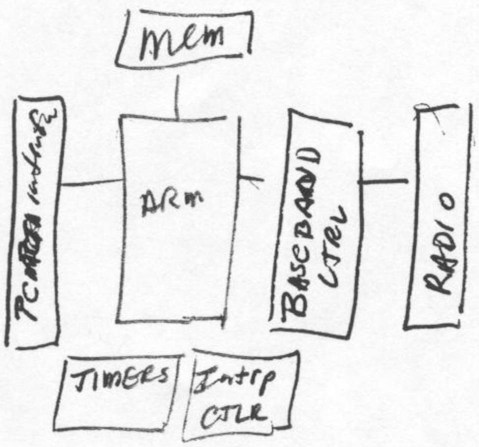
8 ports ⇒ 64 mHz

:

⇒ make routing & scheduling decision on many bits

- Ethernet header
  - IP header
    - Http request

What about wireless



Present fairly conventional dma dq interface
to host

Federation of devices to support low level
operation

Programmable controller to implement
- MAC

sometimes a dedicated baseband controller
- Hopping, sequencing

Discussion

- Physical Parallelism vs Virtual Parallelism
  for concurrency intensive architecture.
- HW support for protocol processing in
  deeply embedded networks
- Perspective on Jason's lecture