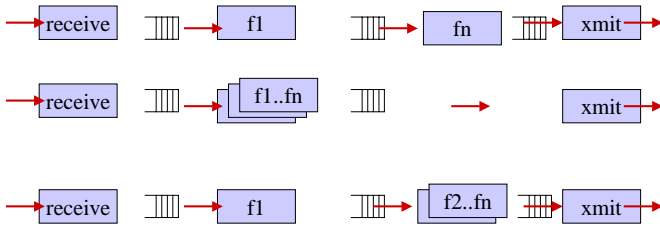


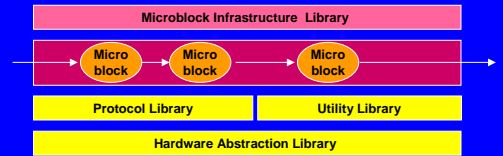
Programming model



issues: size of fi, cyclecount for fi,
state across packets at fi, state for a packet across all fis

Microengine Programming Model

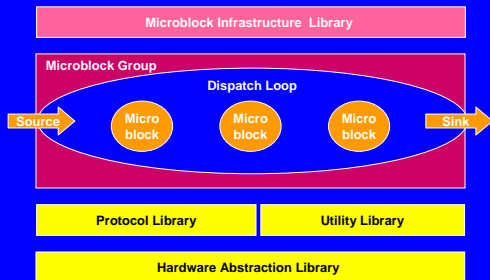
**



** Ibar Bishara's presentation
at IXA Summit 04

Microengine Programming Model

**



- Hardware abstraction
 - OS-like functionality – APIs for memory manipulations (sram, scratch, dram...), buffer/queue management, MSF access...
- Protocol library
 - header field extraction, validation, update for popular protocols (ipv4, Ethernet...)
- Utility library
 - functions for hash table, CAM accesses, threads API...
- Infrastructure library
 - set up application specific packet meta data, pipeline parameters...
- Software framework
 - .../opt-links/ixp/IXA_3.1/ ... /dataplane-library,
 - ... microblocks-library
 - EDU_Wkstn ... sample_application – Radisys stuff
 - SDK_Tools/ src/librabry or me-tools

Programming the IXPs

- Rely on tools (Windows)
 - Software Development Kit – Development Workbench + Simulator (Transactor)
 - cycle accurate simulation (or so)
 - traffic generator
 - build DLLs to implement interaction with special components
 - e.g., custom protocols, interactions with XScale or host...
 - Architecture Development Tool
 - initial design and analysis
 - /net/hp31/ixpdev/exports-sdk4.0/SDK/nassaupr8_noncrypto.zip

Program in

- microC (familiar, portable),
- or
- microcode (efficient, best use of platform features)
- or
- mix of both

Build and run code with

- workbench tools (hardware mode)
- or
- microcode assembler and linker on ilab and XScale command line utilities on ilab-ixp
 - ssh ilabn -> telnet ilabn-ixp1 -> load, start, stop, sram...

Dispatch Loop Structure in Microcode **

```
// include files
#include "dispatch.uc"
#include "dl_system.uc"
#include "IPv4_Fwd.uc"
#include "l2_encap.uc"
#include "l2_classify.uc"

Init#:
// The following code is
// executed once
DL_Source_Init []
IPv4_Fwd_Init []
L2_Classify_Init []
L2_Encap_Init []
DL_Sink_Init []
```

```
// The following loop runs for
// every packet
loop#:
// Get a packet from the Scratch
// Ring
DL_Source []
// Perform Layer-2 Classification
// on the packet
L2_Classify []
// call the IP forwarding block
IPv4_Fwd []
// Perform Layer-2 Encapsulation
L2_Encap []
// pass the packet to next
// Microengine
DL_Sink []
BR [ loop# ]
```

IXA Software Framework **

