Architecture Abstraction – IXP1200

niraj shah
Architecture Abstraction Levels

Domain specific models

Macro behavior

Primitive/Compound interaction

Primitive elements
Primitive elements

- **micro-engine**
  - composed of 4 threads
  - thread scheduler

- **memories**
  - registers
    - GPRs
    - SRAM/DRAM transfer registers (read & write)
    - CSRs
  - Scratchpad
  - SRAM
  - DRAM
  - Specialized
    - RFIFO
    - TFIFO
    - SRAM LIFO registers

- **Communication fabrics**
  - IX Bus: MAC
  - Ready Bus: control info
  - SRAM bus
  - DRAM bus
  - PCI

- **Special purpose HW**
  - hash engine
  - xmit/recv state machines
  - ready bus sequencer (autonomous, micro-coded)

- **Peripherals**
  - MAC
  - ARM
Primitive Interaction

- uEngine0
  - uEngine1
  - uEngine5
- dram
- sram
- scratch
- CSRs
- TFIFO
- RFIFO
- hash

- DRAM bus
- SRAM bus

- Ready Bus
- IX bus

- IX bus i/f log.
- rdy bus seq
- IX bus seq
- Rx state mach
- Tx state mach
- mach
- mach
- mach
- mach
- mach
Primitive Interaction

- **uEngine -- SRAM/DRAM transfer registers**
  - direct access
  - can *only* read from a Read register
- **uEngine – T/RFIFO: via SRAM xfer registers**
- **uEngine – CSRs: via SRAM xfer registers**
- **uEngine -- SRAM LIFO registers: push/pop instructions**

- **thread – uEngine**
  - HW thread scheduler
- **thread – thread**
  - signals

- **TFIFO – MAC: direct via IX bus, controlled by Tx St Mach**
- **MAC – RFIFO: direct via IX bus, controlled by Rx St Mach**
- **DRAM – TFIFO: direct access, initiated by uEngine**
- **CSRs – Rx/Tx St Mach: direct**
- **Ready Bus Seq – CSRs: direct, via Ready Bus**
- **MAC – Ready Bus Seq: direct, via Ready Bus**
Compound Interaction

- **uEngine – uEngine**
  - scratch
  - SRAM
  - DRAM

- **uEngine – MAC:**
  - data: via DRAM & TFIFO
  - ctl: via SRAM xfer reg, CSRs, Rx/Tx St Mach

- **uEngine – Rx/Tx St Mach:** via CSRs

- **CSRs – MAC:** through Ready Bus Seq. & Ready Bus
Macro Behavior

- sharing GPR b/t threads on same uEngine
- memory access + context swap/sig done
- send/recv 64 bytes to/from MAC
- SRAM R-M-W atomic operation
- SRAM/Scratch lock mechanism
- malloc()

```c
void send64bytes() {
    <check port is free using CSR xmit_rdy_lo>

    curTFIFOptr = <CSR xmit_ptr>

    <send 64 bytes from DRAMptr to TFIFO[curTFIFOptr]>

    <write status_word to TFIFO>

    <write CSR xmit_validate>
}
```