

Advanced Computer Systems Architecture

Chip-Multiprocessors: Applications and Architectures

CSE 561M

Prof. Patrick Crowley

Plan for Today

- Questions
 - Please send any questions to me via email
- Today's discussion

Objective

- Discuss some IXP mechanisms from Ch. 6
 - Describe available resources
 - Example code
 - I recommend that you try these out in the simulator
- Discuss projects

IXP Mechanisms

- Hash unit
- CRC unit
- Local memory
- Multiplication
- Random number generation
- Timers & counters

Hash Unit

- Purpose: hw-accelerated hash calculation
- Microengine C

```
hash_in = data;  
hash_128(&hash_in, &hash_out, 1,  
         sig_done, &hash_signal);  
wait_for_all(&hash_signal);
```

- Microengine Assembly

```
hash_128[$hash_data, 1], sig_done(sig)
```

CRC Unit

- Purpose: hw-accelerated CRC (data integrity) calculation
- Can be used for hashing
 - Plus: local to each ME
 - Minus: only 16 or 32 bits
- The `crc` instruction accumulates the CRC value in the `crc_remainder` CSR, and also copies the input data to a specified output register
 - This enables on-the-fly CRC calculations

CRC Microengine Assembly

```
; Assume data to CRC is in $s_transfer_in[3:0]
; Current remainder has been put into gpr_n
; Compute CRC while moving data to d_transfer_out[3:0]
local_csr_wr[crc_remainder, gpr_n] ;restore remainder
nop ; nops could be replaced by useful instructions
nop
nop
crc[crc_ccitt, $$d_trans_out_0, $s_trans_in_0]
nop
crc[crc_ccitt, $$d_trans_out_1, $s_trans_in_1]
nop
crc[crc_ccitt, $$d_trans_out_2, $s_trans_in_2]
nop
crc[crc_ccitt, $$d_trans_out_3, $s_trans_in_3]
nop
nop
nop
nop
nop ; five intervening instructions before reading result
local_csr_rd[crc_remainder] ;get the new remainder
immed[gpr_n, 0x0000]
```

Local memory

- ME-local memory (640 words) is addressed via indexing
 - Two available indices
 - Index registers can be post-inc'd and –dec'd

- Microengine C

```
__declspec(local_mem) int var;
```

- Microengine Assembly

```
local_csr_wr[ACTIVE_LM_ADDR_0, address]  
...  
alu[*1$index0[offset]++, regA, +, regB]
```

Multiplication

- Alu directly supports 8 bit multipliers
 - has incremental support for $24*8$, $16*16$, and $32*32$ (32 or 64 bit results)

- Microengine C

```
Product = multiplicand * multiplier;
```

- Microengine Assembly (32*32 w/ 64b result)

```
mul_step[multiplicand,multiplier], 32x32_start  
mul_step[multiplicand,multiplier], 32x32_step1  
mul_step[multiplicand,multiplier], 32x32_step2  
mul_step[multiplicand,multiplier], 32x32_step3  
mul_step[multiplicand,multiplier], 32x32_step4  
mul_step[dest_low,--], 32x32_last  
mul_step[dest_high,--], 32x32_last2
```

Random Number Generation

- Purpose: hw-accelerated random number generation
 - 32b numbers, sequence repeats after 2^{32} reads
- Microengine C

```
local_csr_wr(local_csr_pseudo_random_number, seed);  
rand_val = local_csr_rd(local_csr_pseudo_random_number);
```

- Microengine Assembly

```
local_csr_wr[PSEUDO_RANDOM_NUMBER, seed]  
local_csr_rd(PSEUDO_RANDOM_NUMBER)  
immed(rand_val_reg, 0)
```

Timestamps

- Purpose: fine-grained access to system clock, increments every 16 cycles
 - Timestamp is 64 bits in length in two CSRs, low and high
 - Can be both written and read

- Microengine C

```
local_csr_wr(local_csr_timestamp_low, 0);  
...  
cycles = local_csr_rd(local_csr_timestamp_low);
```

- Microengine Assembly

```
local_csr_wr[TIMESTAMP_LOW, 0]  
...  
local_csr_rd(TIMESTAMP_LOW)  
immed(cycles_reg, 0)
```

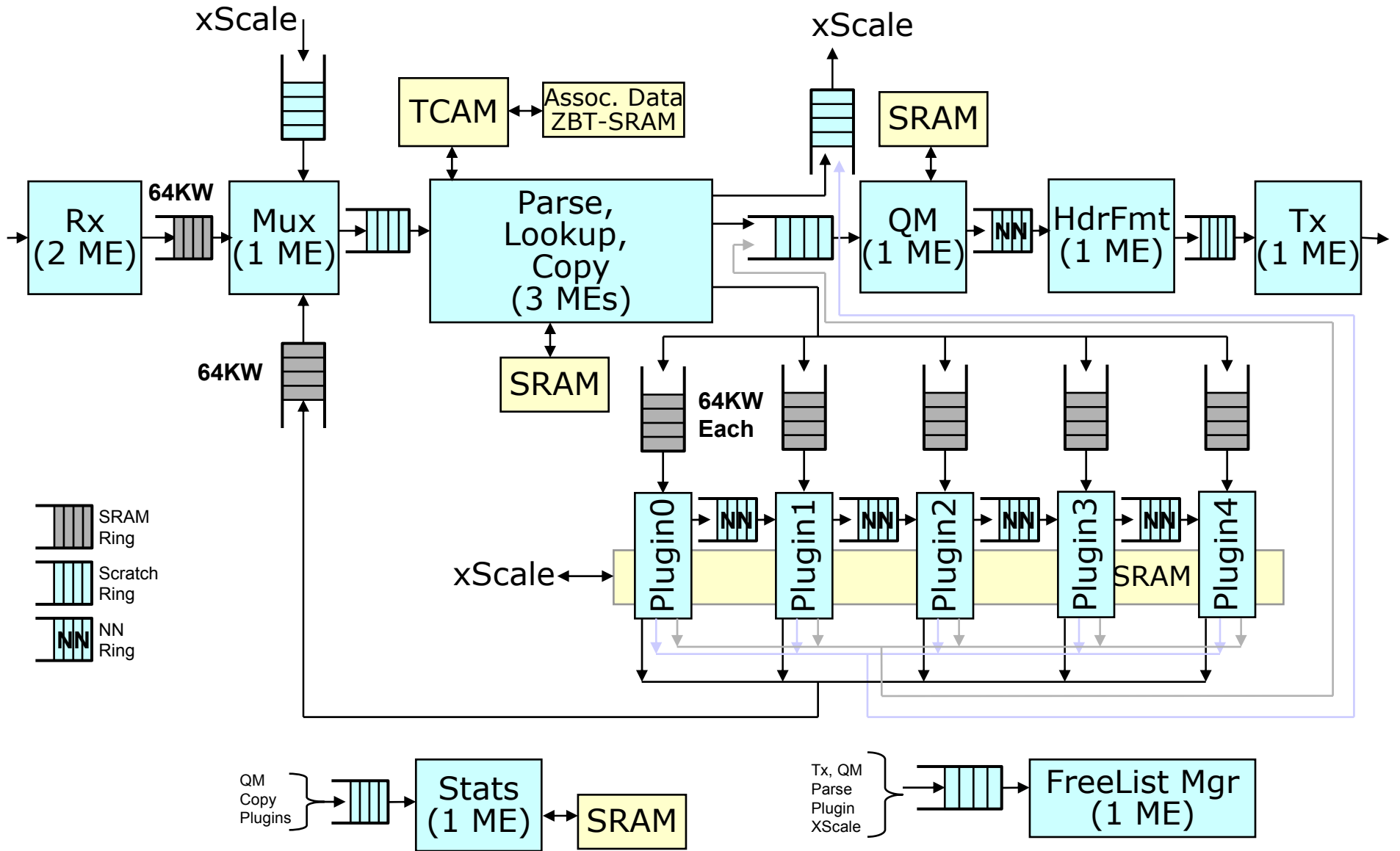
Project Logistics

- Groups of one or two
- We will plan to use NP routers in ONL
- Weekly Milestones
 - Milestone report due each week
 - Submit proposal Feb 26
 - Final presentation Apr 15
- Next week: intro to ONL

Project Milestones

M0	Feb 26	Project Proposal
M1	Mar 4	Design
M2	Mar 18	Implementation 1
M3	Mar 25	Implementation 2
M4	Apr 1	Implementation 3
M5	Apr 8	Wrap-up, Prepare reports
M6	Apr 15	Presentations

ONL NP Router



Potential Project Topics

- Networking
 - Traffic and flow monitor
 - Queue management
 - Protocol implementation
 - SIP, RSVP, Bittorrent
 - IP multicast
 - Packet content search or transformation
 - Extract info from image, video data
- Or other topics of interest to you
 - If you have a special interest, email me *ASAP*

Assignment

- Tuesday
 - J&K: Read Ch. 7
 - **Commentary:** Download the sample `simple_rpt` project and implement a different process step (currently it counts packets seen). In your commentary, describe your processing step and include your code.
- Thursday
 - J&K: Read Ch. 8

Assignment

- Monday, Feb 20
 - J&K: Read Ch. 8
- Monday, Feb 27
 - Project Proposal :
 - Determine group members
 - Propose project idea
 - Use sentences and paragraphs to describe the project
 - 6 week project
 - Break project into milestones (see project logistics slide)
 - Use sentences and paragraphs
 - Demonstrate some understanding of the problem
 - Submit document to newsgroup, and bring hardcopy to class