TWO INSTRUCTION COUNTERS

-A brief survey

EH Susarla
3.25.68
FUNDAMENTAL VIEW OF TWO INSTRUCTION COUNTER MACHINE

* External or architectural definition
  - Second IC is equivalent to second CPU
  - Hence, all questions about logical interlocking and cooperation are to be answered in a multiprocessing sense

* Internal or organizational definition
  - Maximize sharing of facilities
  - Recognize performance degraded on single problem
  - Duplicate facilities only to insure logical integrity
PERFORMANCE ESTIMATE

CPU only

- Net utilization from one IC low-capacity is about 1 op every 1/3 cycle
  actual performance 1 op every 1/2 cycle to every 2 cycles (16% to 66%)

- "Hurry up and wait" effect probably not hurt

- Worst bottleneck is probably eq adder to BIM

- Performance degraded on single problem

- Net effect estimated favorable
### SEGMENT OF LINK EDIT

**Program Branch Limited**

*(Even S Stack 20% Utilize)*

**Low Usage of EA Ports**

*(50%)*

"Hurry-Up-Wait" Form
Limit is program dependencies
(out of order in F not greatly utilized)

High usage of F buffers

Low usage of F facilities
(50% of *+)

Low usage of EA ports

<table>
<thead>
<tr>
<th>GRID ORIGIN</th>
<th>CORNET FOP6 RXB12 MEM4 ADD1/4 MPY5/8,2/5 D62461421452 6 4 861685</th>
</tr>
</thead>
<tbody>
<tr>
<td>S STACK</td>
<td>KKKKKKKLMLNCPQTUVIJUJVUYAZAABCCDEFGHIJKKKKKKLMMPPPPQQQRRRRTUUVVWWYXZZZZZAABBCCDDDEEFMMMMNMNPQTJYVYYYYYZZABCCDEEFGHIJKLMNNNNMPPPPQRRRTTUUVVWVVWXYYYZZABCCDDDEEEFFGGGHHHHKPPPPLPRQTVYBBBBDCEEFGHIIJKLMNQQQQQQQQRSSSSTUVVWVVVWWWXXYYZZAABBCDEEEFFGGGHHHHIIJJ<em><strong><em>AAQT</em>T</strong>**UZC</em>EFCGDJKLMN**<em><em>QPSG3RS</em>YSTUWXXXXYA</em>BDEEEEFFGHHHHIIJJ</td>
</tr>
<tr>
<td>EA ADDER 1</td>
<td><em>A<strong>X</strong>YA</em>CD<em><strong>EHIIIQJ</strong></em>KVL* <em><strong>MMDP</strong></em><strong>OQ</strong>ST**</td>
</tr>
<tr>
<td>EA ADDER 2</td>
<td>RRSWWXXXXX NNNNN SSSSS</td>
</tr>
<tr>
<td>G STACK</td>
<td>SRSWWXXXXX NNNNN SSSSS</td>
</tr>
<tr>
<td>FIXED ADD</td>
<td>SRSWXNSIS</td>
</tr>
<tr>
<td>F STACK</td>
<td>AAAAABBBBHHHHIIIPQPPQQQQQQYTCCCDDDDDDEEEHIIIIIITIJJJJK0000PQQQQQRRRRTRRRTTTT</td>
</tr>
<tr>
<td>FLOAT ADD</td>
<td>ABHUBFEGIMWR</td>
</tr>
<tr>
<td>FLOAT M/D</td>
<td>ZZZZGGGGGGPPPPPVVVVVVCCCCCCHHHHHHLLLLLLVVVQQQQQYYYYCL</td>
</tr>
<tr>
<td>FLOAT L+TST</td>
<td>CKJMZTY</td>
</tr>
<tr>
<td>FLOAT ST</td>
<td>LIAQ</td>
</tr>
<tr>
<td>FL RX BFRS</td>
<td>BBBBAAA99998888777899999999BABBAAAAABBCCCBCCCCCCCBCCCCCCCBCCCCCCCCCBBBAAAB</td>
</tr>
<tr>
<td>FX RX BFRS</td>
<td>111111111111111111111111111111</td>
</tr>
<tr>
<td>SARS</td>
<td>22222222233333333333333333333322222222222222222222222222222222222222222222</td>
</tr>
</tbody>
</table>

**This OP Delayed**

17 Cycles (goes on cycle 85 rather than 68)

20% Loss

EA Utilization ~ 80%

- Priority to CORNET

**Run CORNET?**

**Link Edit Together**

(EO'd only)
CPU with BLM

- With 1 IC, CPU waits when MS-HSS swap needed (about 20 cycles idle)
- With 2 ICs, CPU completely idle only when two swaps needed simultaneously

- Potential harm is each IC disturbs others lines excessively
- Amdahl's model (1967) indicates net effect favorable

Total system

- Need two programs ready in MS to utilize both IC's

- Arnold's model (1968) net effect favorable throughput improvement equivalent to one IC which is 40% faster
TYPICAL ALGORITHM FOR INSTRUCTION SEQUENCING

* Once in S or F or G instructions vie normally (i.e. of those ready, first in, first out)

* Choose next IR to Prefetch to S

Object: Don't let one stream clog the stacks, as it might if waiting for data swap.

* Choose easy way in odd-ball cases usually

  "Pause" in \( \overline{I} \)
  
  "Single cycle" in \( I \)
  
  "Retry backup" in \( I \)
  
  "Wait state" in \( I \)

\( I \) slows also

\( \overline{I} \) single cycles also

\( \overline{I} \) stops temporarily

\( I \) runs normally
COST ESTIMATE

* Second IC adds about 15K circuits in CPU and about 4K circuits in BLM
* Hence about 10% increase
* Half of the circuits are "easy" (registers)
  " " " " " " " " " "hard" (controls)
F UNIT

Current
- 12 64-bit registers
  with full renaming
- start 1 of 6 contending
- adder
  multiplier/divider

Changes
- 12 registers probably suffices
  else increase to 16
  (may hurt adder)
- added control for more
  names, condition codes,
  interrupts, etc
- no change in execution units
Current

- 16  32 bit registers  
  4  "  RX buffers  
  8 K
- start 1 of 4 contenders  
  11 K
- adder
  shift, logic
  divide, convert, etc  
  10 K
- VFL, decimal  
  11.5 K
  \( 40.5 \) K

Changes

- increase to 32 32 bit registers  
  6  "  RX buffers  
  6 K
- added control for more registers,  
  two condition codes, two  
  interrupts, etc  
  1 K
- no change in execution units  
  of VFL  
  0  
  \( 7 K \)
S UNIT

Current
- 6 deep op stack
- Interlocks
- Move 2 ops each to F and G, and bubble up
- Move 2 ops to ea add
- BX buffer assignment, conditional instructions, store in stream, etc.

Changes
- Stack
  - Basic interlocks
  - Select 2 for F/G/ea

- Added control for store in two streams, more registers, etc.

4.5k
2k
4.5k
5k

(18k)

0

3k

(3k)
I UNIT

Current
• 10 64-bit buffers with addresses \(4\frac{1}{2} \times 1\)
• prefetch controls \(2\frac{1}{2} \times 1\)
• predecoder \(4 \times 1\)

Changes
• increase to 12 IB's and share \(1 \times 1\)
• duplicate most of prefetch \(2 \times 1\)
• no change in predecode \(0 \times (3k)\)
MISC

- Clock 7k (no change)
- Maintenance 10% (add 1.5k to 13k)
- Interrupt structure (precise)
  no cost estimate for 1 IC
  cost doubles for 2 IC's

TOTAL for CPU

1 IC  141k
2 IC's  141k + 15.5k
BLM

Current

- Two ports each with 6 levels
- Each accepts 64 bit request for load/store/instr from CPU or IOM
- Two swap start mechanisms
- 32K byte capacity in 512 64-byte lines

Changes

- No basic changes
- Some added control as more return names
- May need more than 6 levels in each stack (8?)

55K

0

1K

(3K?)
IOM

Current  32 channels  43K
No changes

TOTALS

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>△</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>141 K</td>
<td>15 K</td>
</tr>
<tr>
<td>BLM</td>
<td>55 K</td>
<td>4 K</td>
</tr>
<tr>
<td>IOM</td>
<td>43 K</td>
<td></td>
</tr>
<tr>
<td></td>
<td>239 K</td>
<td>20 K</td>
</tr>
</tbody>
</table>
RAS CONSIDERATIONS

* Net effect is unfavorable
  - Second IC adds circuits and complexity
  - Insufficient probability of one IC working when other has failed
    (contrast: two CPUs enhance availability)
* Programming

  - OS multiprogramming support unknown
    SIESTA may/may not make critical assumptions about 65MP or 85MP organizations

* Market

  - Does PRICE of minimal configuration increase?

* ACS Goals

  - Does schedule change?

  - Is second IC within charter?
    prestige, leadership, fallout, etc