A METHODOLOGY OF ASYNCHRONOUS DESIGN: MICROPipeline IMPLEMENTATION FOR QoS-ATM MANAGEMENT

Rubén Alarcón, J.C. Bordoll
Universitat Autònoma de Barcelona .ETSE. Campus U.A.B.
08193 Bellaterra. Spain
e-mail: ralarcon@microelec.uab.es

THIRD ACiD-WG Workshop
Heraklion, Crete, Greece
27-28, January 2003

Motivation for Asynchronous design

- Low power: clocks cause unnecessary activity, clocks cause EMC problems.
- Modularity: Building block approach (Asynchronous Micropipeline with matched delays).
- Performance at system level: ‘typical’ rather than ‘worst-case’.
- ATM protocol: transfers an irregular high-throughput data streams.
- Clock-to-data skew: source-synchronous clocking is inflexible.
- Intrinsic jitter: phase-locked loops introduces more jitter into the system.
ASYNCHRONOUS MICROPipeline WITH MATCHED DELAYS
ANALYSIS OF THE THROUGHPUT

In a stage \((i)\), the minimum cycle time \((P)\) for this stage is the sum of the forward and backward latencies
\((P = L_f + L_r)\):

\[L_f = TQ + T_{logic} + t_u\]
\[L_r = t_{AA}\]

Throughput = \(1/P\)

THE SWITCH ARCHITECTURE

- Arriving cells on all \(M\) inputs ports are stored in the shared memory and organised into separated queues.
- If a new packet arrives and the memory is full, it can be admitted by pushing out a packet with lower priority.
- QoS requirements can be specified through probabilistic values.
QoS-ATM MANAGEMENT ALGORITHM

- An ATM network system must provide a greater number of QoS to improve system efficiency.

- The algorithm for managing QoS is based on the sorting concept.

- In OC-48 transmission the slot time is 170 ns/cell, and in OC-12 is 680 ns/cell.

- The Sorter Unit is based on both serial and parallel architecture.

SORTER UNIT ARCHITECTURE

- The data format has two fields: the memory address field and the priority field.

- The number of serial blocks is according to the number of cells that the shared memory can store.

- Groups, from different queues, are sorted by the parallel sorter according to their priority.
THE SORTER ARCHITECTURES

- The number of serial blocks depend on the number of the serial input vectors.

- The bitonic parallel sorter has 6 micropipeline stages, each stage has a similar logic structure.

DESIGN FLOW

- The Library of Parameterised Modules (LPM) allows easily the portability.

- The VHDL description is based on GENERIC sentences that are used to define parameters.

- The entire circuit and its components are described as a hierarchical project.
DESCRIPTION IN VHDL LANGUAGE

- All data processing blocks are described using simple LPMs.
- C-element is implemented with a simple LPM_add_sub.
- Matched delays are implemented using an equivalent chain of several C-element.

C-ELEMENT BASED ON LPM_ADD_SUB

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
LIBRARY lpm;
USE lpm.lpm_components.ALL;

ENTITY cnotm IS
PORT(
    a : IN STD_LOGIC;
    b : IN STD_LOGIC;
    aclr : IN STD_LOGIC;
    c : OUT STD_LOGIC
);
END cnotm;

ARCHITECTURE xx OF cnotm IS

signal OUT1 : STD_LOGIC;
signal OUT2,OUT3 : STD_LOGIC_VECTOR(0 DOWNTO 0);
BEGIN

prueba1  : lpm_add_sub
GENERIC MAP (LPM_WIDTH => 1,
LPM_REPRESENTATION => "UNSIGNED")
PORT MAP (dataa => OUT3, datab => OUT2, cout => OUT1,
cin =>OUT1);

END xx;

RESULT OF THE IMPLEMENTATION ON FPGAs

- The implementation of prototypes is done in two phases, in the first one, the mapping is made on boards ALTERA FLEX 10K and XILINX XC4010.
- These implementations allow to validate stages of the design methodology as synthesis, layout and timing verification.
- A live circuit test allows to corroborate the results obtained on the above platforms.

<table>
<thead>
<tr>
<th>MAIN BLOCKS (4 BITS / VECTOR)</th>
<th>C ELEMENT</th>
<th>LATCH</th>
<th>CMP</th>
<th>MUX</th>
<th>DELEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVELS OF LOGIC</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NUMBER OF CLBs (XC4010XLPC84)</td>
<td>1/2</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>NUMBER OF LIs (FLEX10K20RC249)</td>
<td>1</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Evaluation of the main blocks

<table>
<thead>
<tr>
<th>SORTER CIRCUITS</th>
<th>AREA ON FPGA</th>
<th>THROUGHPUT MEASUREMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARALLEL FLEX10K20RC240</td>
<td>1021 LIs (88%)</td>
<td>28.93 M outputs/sec (34.60 ns/output)</td>
</tr>
<tr>
<td>SERIAL XC4010XLPC84</td>
<td>336 CLBs (94%)</td>
<td>14.04 M outputs/sec (71.23 ns/output)</td>
</tr>
</tbody>
</table>

Throughput measurements
RESULT OF THE IMPLEMENTATION ON FPGAs

• In the second phase, both sorter circuits serial and parallel are expanded for 8 input vectors; 8 and 16 bits each.

• The implementation on ALTERA APEX 20KE, shows that both sorter circuits can meet the requirements of the Sorter Unit for OC-12 (16 bits) or OC-48 (8 bits) transmission.

<table>
<thead>
<tr>
<th>SERIAL ARCHITECTURE</th>
<th>AREA</th>
<th>THROUGHPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEX 20K100EQC240-1</td>
<td>1610 LEX (35%)</td>
<td>40 M outputs/seg (25 ns/output)</td>
</tr>
<tr>
<td>8 VECTORES 8 BITS / VECTOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APEX 20K100EQC240-2</td>
<td>2409 LEX (57%)</td>
<td>25.3 M outputs/seg (39.8 ns/output)</td>
</tr>
<tr>
<td>8 VECTORES 16 BITS / VECTOR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Throughput in the serial sorter

<table>
<thead>
<tr>
<th>PARALLEL ARCHITECTURE</th>
<th>AREA</th>
<th>THROUGHPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEX 20K100EQC240-1</td>
<td>2860 LEX (68%)</td>
<td>50 M outputs/sec (20 ns/output)</td>
</tr>
<tr>
<td>8 VECTORES 8 BITS / VECTOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APEX 20K160BCEC356-2</td>
<td>4410 LEX (69%)</td>
<td>40 M outputs/sec (25 ns/output)</td>
</tr>
<tr>
<td>8 VECTORES 16 BITS / VECTOR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Throughput in the parallel sorter

PERFORMANCE OF THE SORTER CIRCUITS

A detail of the board test (parallel sorter)

A detail of the board test (serial sorter)
CONCLUSIONS

• The use of asynchronous techniques in communication circuits, apart from other intrinsic advantages, can avoid or at least minimise jitter, skew and synchronisation problems.

• In QoS-ATM, the asynchronous micropipeline allows considering the resources for typical case conditions.

• The description VHDL-LPM, allows the circuit maintains technological independence, sacrificing minimum silicon efficiency.

• To validate the presented approach, we showed the modular asynchronous micropipeline implementation of a Sorter Unit.