A DSP ASIC Design Flow Based on VHDL and ASIC-emulation
Jan Andersson, Ericsson Microwave Systems AB, Mölndal, Sweden

Abstract

In an ASIC project VHDL-simulation, VHDL-synthesis and ASIC-emulation has been used. The project is completed and described in this paper.

As an introduction a short presentation of the end product, MINI-LINK, is given.

The project started with a rough design flow. The characteristics of the ASIC and its target formed the details as time went by. The ASIC, its target and the final design flow is presented.

One step in the design flow is verification by ASIC-emulation. What ASIC-emulation is, why it was chosen and how it was used in the project will be described.

The paper is concluded with a summary of the results.

1: Introduction

MINI-LINK is a family of radio links from Ericsson Microwave Systems AB.

In order to improve the link a DSP-ASIC has been designed. The design flow described in this paper has been used.

2: MINI-LINK

MINI-LINK is a communication medium based on microwaves. It carries information from point A to point B and is an alternative to copper and fibre cables. The information carried is usually telephone calls.

Figure 1 MINI-LINK [1]

One application is cellular mobile telephone networks where MINI-LINK connects radiobase stations to switching centres.

Figure 2 Cellular mobile telephone network

MINI-LINK has a transmission capacity of 2 Mbps and 8*2 Mbps and a range of up to 30 km or more[1].
3: The ASIC and its target

As mentioned in previous chapter a radio link is an alternative to copper and optical fibers. Figure 3 shows a radio link equivalent to a copper wire. The wire is replaced by base band units, micro wave units, antennas and microwaves. The DSPASIC is located inside the base band unit (target).

![Diagram](image)

BBU=Base Band Unit; MWU= Micro Wave Unit

Figure 3 Radio link

The DSP-ASIC is used adjacent to and in interaction with an analog base band system.

On the receiver side it closes a number of analog loops. Both the digital part and the analog part do signal processing.

![Diagram](image)

ADC=Analog to digital converter; RX=Receiver
DAC=Digital to analog converter; TX=Transmitt

Figure 4 Base Band Unit - target

It performs calculations such as multiplications $\arctan(x/y)$, comparisons and additions on each positive clockedge.

The size of the DSP-ASIC is 50000 Gates.

4: Design flow

The project started in the beginning of February 1994. No design flow was readily available at department A at that time.

One important issue was time and how to ensure the schedule. The goal was to have the ASIC verified at system level by the end of december 1994.

4.1 A tentative beginning

The first question was how to verify the design on system level. Verification on system level is in this context a measurement of BER (Bit Error Rate) as a function of SNR (Signal to Noise Ratio).

Was it enough to make a coarse system verification by simulation early in the design phase and then a full system verification when an ASIC finally became available? No, the probability of introducing an error somewhere in the design flow or not to cover every aspect in the system design was believed to be too high.

In order to make an early and full system verification with real hardware it was decided that a FPGA-version of the ASIC had to be made as soon as possible.

In order to make the design as independent of the technology as possible, VHDL was chosen as the design language. The plan was to map the VHDL-design on a FPGA-based board and on the ASIC at the same time and when the FPGA system verification was done start the ASIC lay out.

The ASIC was partitioned into eleven modules. The modules was to be designed by the use of VHDL and interconnected by a circuit diagram.

It was not long until the design flow had to be changed...
4.2 The project grows

In March, one month after the start of the project, it became evident that the project team was not big enough to carry out the task.

The design flow was then partitioned into one technology independent and one technology dependent phase. In the first phase the ASIC was specified and simulated using VHDL and in the second phase the code was synthesized and mapped onto a vendors library. The second phase also included scan insertion, ATPG and I/O-issues.

Department B which had experience with the technology dependent phase from earlier projects agreed to do this phase.

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>TECHNOLOGY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department A</td>
<td>Department B</td>
</tr>
<tr>
<td>System design</td>
<td></td>
</tr>
<tr>
<td>VHDL-SPEC</td>
<td>-Design errors</td>
</tr>
<tr>
<td></td>
<td>-Synthesis errors</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REALISATION</td>
</tr>
<tr>
<td></td>
<td>-Synthesis + simulation</td>
</tr>
<tr>
<td></td>
<td>-Scan insertion</td>
</tr>
<tr>
<td></td>
<td>-ATPG</td>
</tr>
</tbody>
</table>

Figur 6 Dividing work load between departments

4.3 Planning the FPGA based board

Due to the tightly scheduled project planning, design and verification had to be done concurrently.

In April 1994, two months after the project start, the size of the design was estimated to 40000 gates.

Concurrent with system and VHDL-design a FPGA based board had to be designed. The strategy was to use the circuit diagram describing the top level as the board schematic and map each VHDL-module onto one or two FPGA:s.

The number of FPGA:s was estimated to seventeen. It became evident that the FPGA approach was a project in itself.

Since this had to be done concurrently with the VHDL-design a problem had been identified.

4.4: Changing verification method

It became evident that the FPGA based board was too complex and too large to fit into the project. Alternatives were discussed. Was it possible to rely on system simulation and wait until the ASIC was available? Again the answer was no. ASIC-emulation was proposed as an alternative. Since this was something new and unknown for department A it was decided that this alternative should be investigated.

It became obvious that ASIC-emulation could be used to solve the problem. At least in theory. ASIC-emulation is described in more detail in chapter 5.

The result of the investigation was that the FPGA based board was replaced with an ASIC-emulator.

The ASIC-emulator was delivered in June 1994.

4.5: System and VHDL-design

Concurrent with the preparation for system verification different parts of the design flow were developed by individuals in the team.

System design and verification of algorithms was made by the use of a simulation program written in BASIC. Specifications where handed over to a VHDL-designer and a VHDL-verifier.

In order to ensure the transformation from module specification to VHDL-model, testvectors from system simulations were used. The result was reported and reviewed by the system designer.

```markdown
<table>
<thead>
<tr>
<th>TEST VECTORS</th>
<th>REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG</td>
<td>MATHCAD</td>
</tr>
<tr>
<td>DUT</td>
<td></td>
</tr>
<tr>
<td>TEST BENCH</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 7 Feeding back module behaviour
4.6 Testcase design

The strategy was to make a VHDL-test bench for each VHDL-module and deliver the module and the test bench as a simulated and self contained testcase to department B.

A test board containing the analog system was designed. It had connectors to make it possible to connect the ASIC-emulator.

In September 1994 the system verification system was ready to receive an emulated ASIC.

4.8: Verifying with the emulated ASIC

An netlist containing the whole design was delivered from department B in October 1994.

A lot of energy had to be spent on solving minor problems such as:

- Netlist errors
- Own test software development
- Bug in the Emulation software
- Tuning the analog system
- Error reports from the emulator. Timing violations.

In order to trace faults it was possible to record vectors inside the emulated ASIC and use the vectors in the system simulator.

In the end of November 1994 it was possible to transmit information through the system and in the end of December 1994 the system behaved as predicted in the system simulations.

Two design errors was found due to the ASIC-emulation.

One (known) error was not found. The test did not cover the particular signal. The error was found (too late) by coincidence when discussing the design. This was solved with software.

In middle of December 1994 the confidence in the design was high enough to enable a preliminary design release (first sign of).

In January 1995 the preliminary design release was made.

In February 1995 the critical design release (second sign of) was made.

While waiting for the ASIC to be produced more measurements on the system were made.

In March 1995 the real ASIC was received....

4.9: Verifying with the real ASIC

Since the emulated ASIC and the real ASIC had different pin out it was not possible to mount the real ASIC on the test board.

There was two ways to solve this. One is to make a new test board. This was the first approach. But since this new board had to be debugged in order to make it work this approach was abandoned. No debug resources were available.
The other approach was to make a small board that was pin compatible with the emulated ASIC. On that small board the real ASIC was mounted.

In the beginning of March 1995 the ASIC-emulator was removed and replaced by the small real ASIC board. Two hours later it was possible to transmitt information through the system. Happy smiles.

But the system did not perform exactly as the ASIC-emulator. There was a bit error. It took another three weeks to isolate the cause. The A/D-converter on the test board was disturbed by noise.

In the end of March 1995 the system once again behaved as predicted in the system simulations. This time with a real ASIC.

### 4.10: Final Design flow

In previous chapter sub design flows has been described. It was impossible to predict every problem. They were solved as they occurred. Finally there was an working design flow for this particular project. Figure 10 shows the final design flow.

Figure 10 Final design flow
5: ASIC emulation

An emulated ASIC is an ASIC netlist mapped onto physical gates and flipflops, just as an ordinary ASIC. The difference is that the connections are programmable. No masks are needed. The ASIC-emulator is also slower.

It is possible to add probes into the design. The probed signals can then be recorded and viewed on a display.

5.1: Internal function

The ASIC-emulator is a unit consisting of a large number of FPGA:s connected together. Each FPGA has a number of programmable blocks.

The emulation software first divides the ASIC netlist into pieces that will fit into an individual FPGA. It then converts each piece into configurable logic blocks and does a place-and-route operation to map the logic into one of the FPGA:s.

5.2: ASIC-emulator vs FPGA

The figures in the table below is the figures upon which the decision to use ASIC-emulation was taken.

<table>
<thead>
<tr>
<th>Structure</th>
<th>FPGA</th>
<th>ASIC-emulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>40000 Gates</td>
<td>60000 Gates</td>
</tr>
<tr>
<td>Vendor</td>
<td>Consultant</td>
<td>QUICKTURN</td>
</tr>
<tr>
<td>Cost</td>
<td>2.8 Cost units</td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>Vacancy</td>
<td>Own personnel</td>
</tr>
<tr>
<td>Frequency</td>
<td>2 MHz</td>
<td>1 MHz</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery</td>
<td>8 weeks</td>
<td>8 weeks</td>
</tr>
<tr>
<td>Programming</td>
<td>2 weeks</td>
<td>2 days after 3 weeks of prep</td>
</tr>
<tr>
<td>Diagnostic.</td>
<td>&gt; 1 week/error</td>
<td>1 day/$error$</td>
</tr>
<tr>
<td>Reprogr.</td>
<td>2 days</td>
<td>1 day</td>
</tr>
<tr>
<td>Education</td>
<td>Own</td>
<td>2*4 days at vendors site</td>
</tr>
<tr>
<td>Risk</td>
<td>Big design, feasible??</td>
<td>Too big BER due to low frequency</td>
</tr>
<tr>
<td>Advantage</td>
<td>Known method</td>
<td>Product, Debugged SW internal signals observable, Reusable,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disadvantage</td>
<td>Hard to observe</td>
<td>Expensive,</td>
</tr>
<tr>
<td></td>
<td>internal signals, own design</td>
<td>1 MHz,</td>
</tr>
</tbody>
</table>

Table 1 FPGA vs ASIC-emulator

The decision was not obvious. But it was believed that the ASIC-emulator was the safest way to ensure the schedule.

The decision taken was to decrease the frequency of the test target board from 2 MHz to 1 MHz and to buy the ASIC-emulator.

6: Conclusions

- It is possible to move a VHDL design between design systems.
- If ASIC-emulation is used for system verification the test system will be debugged and personnel will gain experience. System verification with the real ASIC will be a lot easier.

7: Acknowledgements

Contributors to or participants in the design flow described has been:

Claes Andersson
Toni Danielsen
Lars Danielsson
Rob Galuszka (QUICKTURN)
Lena Hansson
Tony Holmes (QUICKTURN)
Stefan Jigsved
Robert Lindgren
Dag Lundström
Lars Moeschlin
Lennart Nylén
P-A Thorsén (FRONTEC)
Jan Sandberg
Dan Weinholt
Kenneth Östberg (CVD)
Magnus Österholm (CVD)

A special thank to Brita Nyberg for valuable comments on this paper.

8: References

1. ERICSSON REVIEW No. 4, 1993,
   Dag Jungefelt