A FORMANT SPEECH SYNTHESISER ASIC: IMPLEMENTATION

Marwan A Jabri†, Kiang Ooi Tan‡ and Clive D Summerfield†

†Sydney University Electrical Engineering
Australia
‡Department of Electrical Engineering
University of Edinburgh
Scotland

ABSTRACT- This paper is the second of two companion papers on the design and implementation of a multi-channel formant speech synthesiser Application Specific Integrated Circuit (ASIC). In this paper, we describe the implementation aspects of the project. The use of the silicon compiler FIRST in the conception of the synthesiser has reduced the design time considerably. However, as the 5μm NMOS primitive library of FIRST falls short in providing sufficient processing bandwidth for multi-channel operation, we implemented a new primitive library using the European Silicon Structure (ES2) standard cell design tool (2 μm CMOS). The library is implemented using the MODEL hardware description language. The implementation of the primitives using MODEL is discussed in detail together with the clocking and data synchronisation strategies necessary for reliable high speed bit-serial operation.

INTRODUCTION

With the rapid advances in microelectronics technology, electronic designers are able to migrate large computing systems on a single Very Large Scale Integration (VLSI) chip. As the transitor count of such a chip may exceed hundreds of thousands, designers are faced with a serious complexity problem. Automation efforts of the design process have led to the production of "silicon compilers" and other Computer Aided Design (CAD) tools that aim at speeding up and simplifying the implementation procedure.

A silicon compiler is a computer program that produce the mask layout of a chip from a structural (or behavioural) description of a circuit. The mask layout defines the geometries of semiconductors and conductors "layers" that make up the devices and the wiring elements of a chip. The mask layout is the final product in the design stage of a chip, before its fabrication.

Silicon compilers produce mask layout that corresponds to a well defined integrated circuit technology and fabrication process. Of interest to us are N channel Metal Oxide Semiconductor (NMOS) and Complementary Metal Oxide Semiconductor (CMOS). The success of the implementation of a chip using any of these technologies is subject to strict adhesion to a set of "design rules" determined by the resolution of the fabrication process. Design rules include rules govening distances between layers to avoid any contact during fabrication, the size of contacts between layers, *etc.* With advances in fabrication processes, resolution is improved and layers may be placed closer to each others. This lead to the subject of the "scalability" of a design process, that is, to be able to express a chip mask geometries using a constant (usually called lambda λ) and to give this constant a smaller value as technology improves. Smaller λ means larger scale integration. Unfortunately, scalability of a fabrication process is not a well controlled factor in that, a circuit may need to be redesigned to implement it in a the same technology but with a smaller λ.

In this paper we describe the implementation, on a single chip, of a Multi-channel Formant Speech Synthesiser (MFSS) circuit. The investigation into architecture that meet the sepcifications of the chip was done using FIRST, a bit-serial silicon compiler package and its behavioural simulator, SIMFIRST. Limitations, however, of FIRST to a 5μ NMOS technology were serious. As well as compromising the specifications of the circuit, 5μ NMOS represents an obsolete fabrication technology. This has led us to design a new technology independent primitive library for FIRST that may be used in conjunction with the behavioural simulator. The library has been implemented using the Hardware Description Language (HDL) MODEL on a Solo-1000 system (produced by ES2). The system, including the library, was then used to implement the MFSS chip in a graphics mouse-based computer environment.

BIT-SERIAL COMPUTATION

One of the main tasks in the design of a VLSI chip is to minimise the total area of silicon occupied by the circuit. Beside the obvious advantage of being able to put a large system on a single die, the minimisation of a chip silicon area has also the advantages of reducing the fabrication cost, increasing the fabrication yield, reducing the power consumption and improving its efficiency. Note that in a VLSI chip, much of the silicon area is used by the interconnection wires (data buses, clock signals and other wiring) between sub-blocks rather than by the devices (*e.g.* transistors) themselves. Bit-serial computation is a common technique used to produce VLSI chips that are highly efficient in terms of area, power consumption and speed. In this technique, data buses are reduced to single lines and computation is performed serially on each bit of the data. This has the obvious effect of reducing the area used by data buses and therefore much of the total area of the chip.

As mentioned earlier, most silicon compilers are usually tied to a particular technology: NMOS, CMOS, *etc.* In addition, most of these compilers are also dependent on the design rules of a particular technology. A compiler that has been designed for a 3μ NMOS process will be obsolete for a 1.5μ NMOS process. This is a serious problem considering the present rapid evolution of the microelectronic technology.

THE FIRST SILICON COMPILER

The "FIRST" silicon compiler was designed by Peter Denyer and David Renshaw at the department of Electrical Engineering, the University of Edinburgh. FIRST is a structural to layout compiler. The input to FIRST is a net-list description with arithmetic and other primitive operators. These are used as circuit elements to build more complex systems. Provided with FIRST is the SIMFIRST behavioural simulator. This enables the logic and behaviour of the circuit to be verified, and represents an important exploration tool for early quick circuit investigation.

LIMITATIONS OF FIRST

The Multi-channel formant speech synthesiser hs been exclusively designed using the FIRST Silicon Compiler. FIRST was selcted for this task as it offered a number of advantages. As it was developed for the implementation of fully synchrous bit-serial signal processors it is ideally suited to this task. The SIMFIRST behavioural simulator provides a quick and effective mechanism for circuit verification and acoustical evaluation of the synthesis structure. FIRST is free to research institutions.

Although FIRST, with its NMOS library, was suitable for initial research and development formant speech synthesis structures, it is unsuitable for the implementation of a multi-channel synthesis device and we needed to look at a more advanced MOS process and at methods of upgrading the primitive library.

CHOICES FOR FIRST UPGRADE

Three solutions were possible to upgrade FIRST to support new technologies:

1. to build a full custom based library with hand layout style masks,
2. to build a full custom based library with a symbolic style circuit description, or
3. to build a semi-custom standard cells based library.

Although the first solution may produce the most efficient layout, its cost, and limitations (*e.g.* the library will be tied again to another technology as it is built at the mask level) do not justify its development exclusively for the circuits we intend to build (prototypes and research oriented chips).

The second solution offers a greater degree of technological independence as primitive circuits are described at the symbolic level (gate level) independently of the design rules. The symbolic description is then "compacted" and the mask layout is therefore produced. There are several drawbacks to this solution, however. Firstly, although it is scalable (independent of design rules) it is still technology dependent. For example, the symbolic descriptions of CMOS are different of the those on NMOS. Even, within the same technology, processes may have access to different number of metal layers, which may mean that a symbolic description with two metal layers is obsolete for a one metal layer process. Secondly, the symbolic based solution relies on the presence of an ideal compactor and ideal placement and routing tools. Our experience in compaction, placement and routing indicates to us that such ideal compactors and supporting design tools are not yet at this stage.

The semi-custom solution offers the highest level of technological independence at the cost of an acceptably reduced efficiency (chip area and speed) compared with the two other solutions However, for the type of chips we are aiming to produce, this tradeoff is acceptable and, hence, the semi-custom approach was selected for the implementation of the MFSS chip.

A TECHNOLOGY INDEPENDENT BIT-SERIAL SILICON COMPILER

A technology independent bit-serial primitive library has been implemented, simulated, integrated within the Solo-1000 system and used to successfully design the MFSS chip. The library implementation efforts concentrated on the following phases:

1. structure of overall system,
2. elaboration of a clocking and data synchronisation strategies, and
3. design and simulation of primitives,

These phases are discussed in details in the following sections.

Overall System Structure

Figure 1 shows the structure of the overall bit-serial system configuration. The bit-serial compiler consists of the MODEL HDL compiler, a new bit-serial primitives library, a new symbol (icon) library of the FIRST primitives, the standard cell logic library and other tool-kits provided by the Solo-1000 system. Note that the primitives, NANDs, NORs, *etc*, of the SOLO-1000 logic library have been used to implement the bit-serial library. The input to the bit-serial compiler is in one of two forms: a MODEL

equivalent of the FIRST description or in a graphic form using the schematic capture facility provided by the Solo-1000 system and the new primitive symbol library. Most of the FIRST primitives (see (Denyer & Renshaw 1985) for a complete listing) have been implemented. The remaing primitives are currently being designed. Similarly with FIRST, clock signals are implicit in the library. The designer does not have to worry about providing pads for the clocks and the power and ground signals. In the text input mode, the designer has to convert his FIRST description into a MODEL description which is similar. In the graphics input mode, the designer may use the library of symbols representing the FIRST primitives to design the circuit and then automatically produce the MODEL code using an option of the Solo-1000 system. The symbol library has been built in a way, that the generated MODEL code matches with the FIRST primitive definitions in the bit-serial library.

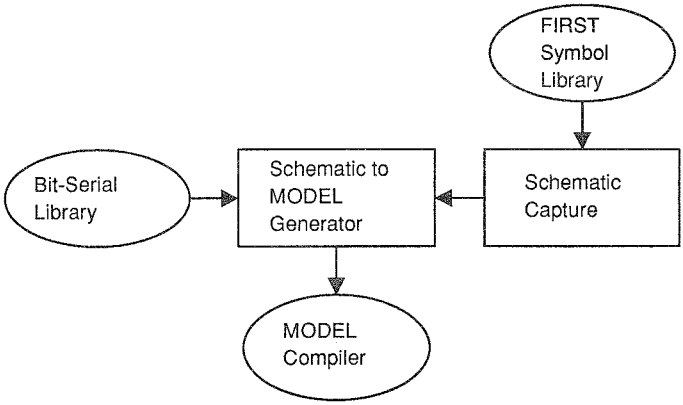

Figure 1: Block diagram of the bit-serial configuration.

Clocking and Data Synchronisation

An important aspect of the overall design of the bit-serial compiler has been the elaboration of its clocking and synchronisation strategy. The compiler has to produce synchronous systems. At first, a single phase clock scheme was adopted. However, this has demonstrated to be unsafe and made the system dependent on the technology through the requirements imposed on the clock by the different FIRST primitives. We then resorted to a two phase clocks scheme phi1 and phi2, described in (Mead & Conway 1980, Denyer & Renshaw 1985). The general clocking and data synchronisation strategy was then based upon the following: data is clocked in any FIRST primitive during phi1 and clocked out of the primtive (and then data is made ready for the next primitive connected to its output) during phi2. A block diagram demonstrating this scheme is shown in Figure 2. This clocking and synchronisation scheme demonstrated to be the safest and the most robust.

Primitives Design and Simulation

The D flip-flop latch shown in Figure 3 was used in the design of most primitives. As it may be seen from the figure, this latch implements the clocking and synchronisation strategy discussed in the previous section. The latch has been built using combinatorial logic gates provided by the Solo-1000 library. This latch was simulated and successfully worked at 40 MHz.

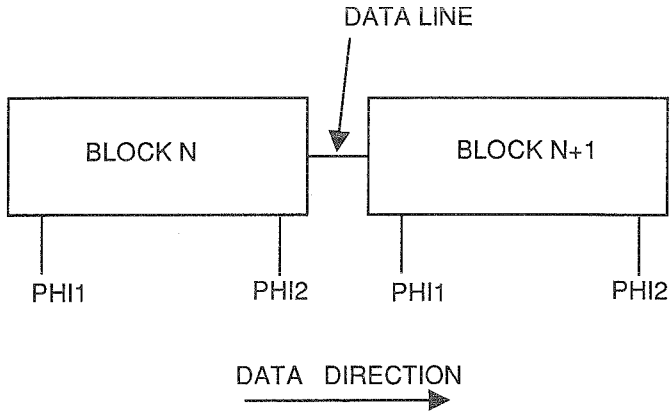

Figure 2: Data clocking and synchronisation.

The other primitives were designed using this latch and simulated at a 35 MHz frequency. All of the simulations were carried out using EXERT, the Solo-1000 switch level simulator.

IMPLEMENTATION OF THE FORMANT SYNTHESISER CHIP

Following the implementation of the FIRST library and the graphic symbol definitions, we implemented the MFSS chip and we generated its MODEL description. Then, the overall chip simulation was started. Note that, as no behaviourial simulator (nor a logic simulator) was provided with Solo-1000, we resorted to the use of EXERT. This was a bit tedious as the circuit included some off-chip long delay elements whose data were to be used in the simulation. We were unable to put these delays elements on chip for the whole sake of simulation, as the MODEL compiler is limited to around 64K transistors. To solve this problem we used the trace output of the delay elements output nodes as produced by the SIMFIRST behavioural simulator, as input to the EXERT simulation. A 'C' program was written to translate SIMFIRST files into EXERT simulation vectors files. An important error was detected during the simulation: We forgot to include a delay element in the I/O pads. The error was rectified, and the chip simulation produced the expected results. The chip was certified and sent to fabrication.

CONCLUSIONS

The design and implementation of a standard cells based bit-serial compiler has been presented. In addition, the implementation of the MFSS has been described. The compiler is technology independent. It is based on a simple but useful idea. Current work is underway to automate the FIRST to MODEL translation. This will lead to a fully correct-by-contruction technology independent bit-serial compiler based on the FIRST language and supported by the SIMFIRST behavioural simulator.

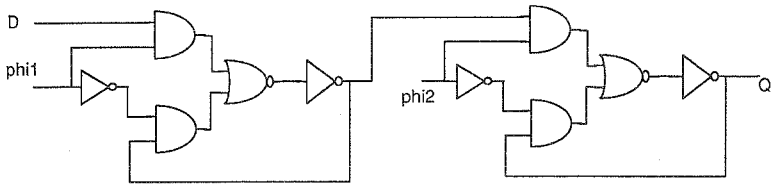

Figure 3: The D Flip-Flop schematic diagram.

ACKNOWLEDGEMENTS

This work is funded by the Overseas Telecommunications Commission (Australia). Acknowledgements to Mr. Mike McGregor at the Department of Electrical Engineering, University of Edinburgh for his assistance with making the ES2 design tools available for this research. Bryan O'Connell has implemented early versions of some of the bit-serial primitives.

REFERENCES

Peter Denyer & David Renshaw (1985) *VLSI signal processing: a bit-serial approach*, Addison-Wesley Publishing Company.

Carver Mead & Lynn Conway (1980) *Introduction to VLSI Systems*, Addison-Wesley Publishing Company.