HIERARCHICAL VHDL LIBRARIES FOR DSP ASIC DESIGN

John V McCanny[1], *Douglas Ridge*[2],*Yi Hu*[2], *Jill Hunter*[1]

1 - Department of Electrical and Electronic Engineering (j.mccanny@ee.qub.ac.uk),
The Queen's University of Belfast, BELFAST, BT9 5AH, Northern Ireland.
2 - Integrated Silicon Systems Limited, 29 Chlorine Gardens (doug@iss-dsp.com),
BELFAST, BT9 5DL, Northern Ireland

ABSTRACT

Methods are presented for the rapid design of DSP ASICs based on the use of hierarchical VHDL libraries. These are portable across many silicon foundries and allow complex DSP silicon systems to be developed in a fraction of the time normally required. Resulting designs are highly competitive with ones created using conventional methods. The approach is illustrated by its application to ADPCM codec and DCT cores.

1 Introduction

The increasing complexity of modern integrated circuits, coupled with ever decreasing times-to-market, underlines the need to further develop efficient silicon system design methodologies. This paper describes methods for the rapid design and implementation of complex DSP ASICs and DSP ASIC cores. These are based on the use of a series of hierarchical, synthesisable VHDL libraries. Blocks within these libraries are parameterised and incorporate a variety of architectural templates, to cover a broad range of application areas. The resulting algorithm-to-architecture-to-chip design flow mirrors in hardware methods used for DSP software algorithm development. Designs produced compare very favourably with ones developed using more conventional methods.

2. Library Structure

The basic philosophy which underlies the approach adopted is that, whilst from a user point of view, many DSP applications vary widely, experience [1-3] has shown that many typical blocks required in such applications can be constructed from a finite library of pre-designed, parameterised (e.g. in terms of wordlengths) templates. These, in turn, can be used to construct higher level blocks (e.g. arithmetic blocks) which can then be used to construct functions such as FFT devices, FIR filters etc. This approach provides an efficient means for capturing silicon "intellectual property" in a manner (a) which allows non-specialists to rapidly create advanced DSP chips and (b) allows more experienced silicon systems engineers to tailor designs to required specifications.

The hierarchical VHDL library structure developed comprises five sub-libraries, as shown in figure 1, with examples of the typical contents documented in the appropriate tables. As illustrated in figure 1, four of the libraries are DSP specific, whilst the fifth (support devices) is included to provide interfaces to other external non-DSP components. In many instances blocks in the higher level libraries have been created using those from a lower level. This means that the optimisation of structures at the lower levels is carried through to higher level functions and results in designs with very high performance, low silicon area and low power consumption.

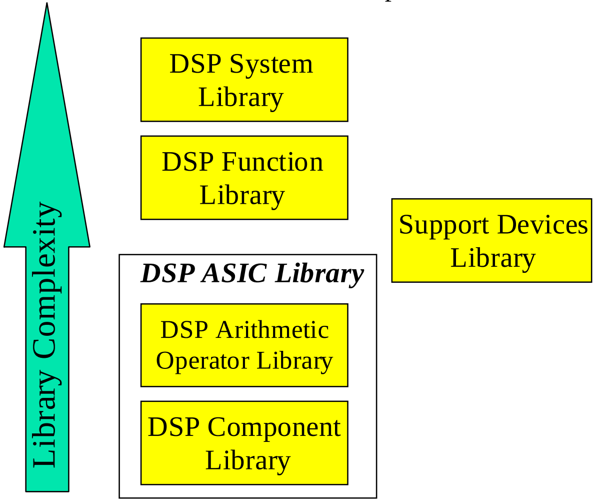

Figure 1. Library hierarchy

DSP Component Library This contains over 70 blocks covering the range of word and data formats described in section 3. Only a limited degree of parameterisation is needed e.g. arithmetic type, word length or counter size.

Component library contents
Counters
Comparators
Delay elements
Data formatters
Data converters
Memories
Shifters

DSP Arithmetic Library The Arithmetic Operator library contains operators such as multipliers, adders, dividers and square root processors. It comprises over 100 optimised blocks. In this case typical parameterisation is in terms of word-length and different types of arithmetic structures used. Examples include ones based on carry-save, carry-look-ahead, carry-ripple and Wallace tree adders. Numerous architectural variants of each block are also incorporated to cover the range of arithmetic and data word formats required.

Arithmetic operator library
Multipliers
Accumulators
MAC cells
Square root operators
Dividers
Adders

DSP Function Library This has a wide range of application. Here a variety of parameters are typically used with the details being function specific. Examples of such parameters are given below.

Parameter
Data word length
Data word format
Level of Truncation
Level of Pipelining
Filter taps/transform size etc.

Function library contents
FIR filters
IIR filters
DFE
DCT
FFT
Median filters
LMS filters
Reed-Solomon
Viterbi decoders
VQ

These allow a high degree of flexibility in tailoring designs to performance requirements. In particular, they allow numerous design alternatives to be readily explored. For instance, it is possible to rapidly create many parameterisable FIR and IIR filter circuit types using different permutations of multiplier/accumulators contained in the arithmetic operator library. On the other hand, it is possible to directly use one or other of the pre-defined library filter functions, in which case, such details are transparent to the user. Typical blocks at this level include FIR and IIR filters, median filters, DCT and FFT processors.

DSP System Library This exists at the highest level in hierarchy and contains complete system level blocks. Here parameterisation allows different versions to be created and, where appropriate, allows the use of different wordlengths. For example, the ADPCM CODEC allows variants to be produced in which the number of duplex channels required can be changed through the supply of appropriate values (see section 4).

System library
MPEG*
JPEG*
H.261*
ADPCM
PRML
Object detection/recognition
**under development*

Support devices library contents
Memory controllers
Microprocessor interfaces
Bus interface controllers
Graphics controllers

3. Data Word Formats

The type of arithmetic used across many different DSP applications can vary considerably. A broad degree of flexibility has been incorporated through the provision of blocks with different forms of arithmetic. These include both floating and fixed point versions. Each word format in this table is capable of being transmitted and processed in a bit parallel, bit serial or digit serial manner. Bit parallel data tends to give the highest performance, at the expense of silicon area and is useful for high bandwidth applications; bit serial data enables reductions in silicon area for low bandwidth applications; digit serial versions provide flexibility and allow the designer freedom in the matching of bandwidth with circuit architecture.

Data Word formats
IEEE floating point
Custom floating point
Unsigned binary
Signed binary sign-magnitude
Signed binary positive-negative
Two’s complement

Two floating point formats have been included. These are the IEEE standard floating point number format and a custom ISS format. The ISS format provides the same

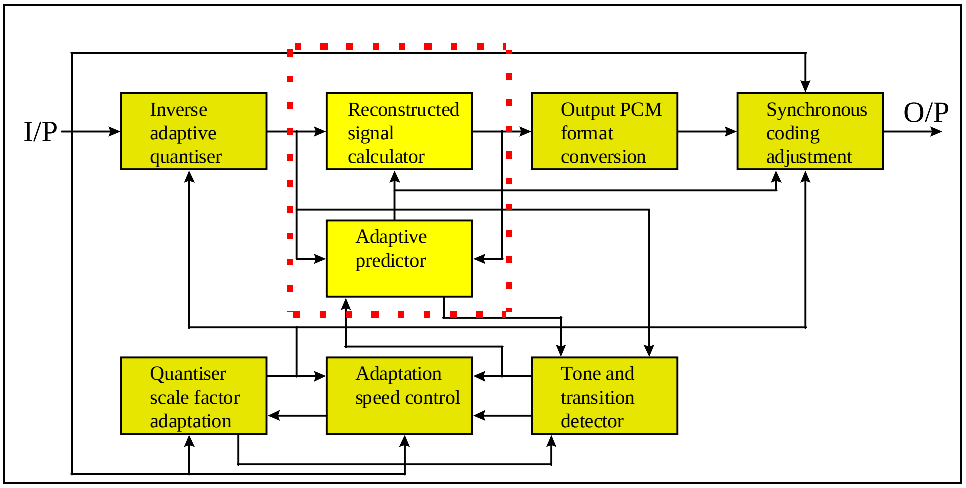
I/P
Inverse adaptive quantiser
Reconstructed signal calculator
Output PCM format conversion
Synchronous coding adjustment
O/P
Adaptive predictor
Quantiser scale factor adaptation
Adaptation speed control
Tone and transition detector

Figure 2. ADPCM encoder block

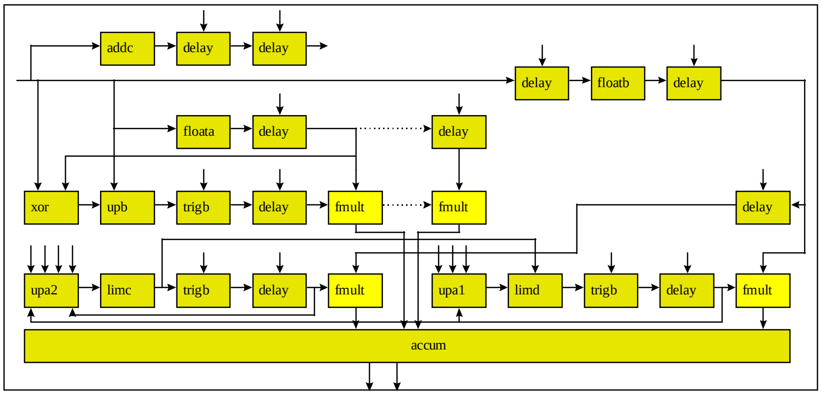

Figure 3. Lower level ADPCM block

accuracy, but enables the development of functions using less silicon area and lower power consumption. Other formats are based on fixed point arithmetic with numerous variants available. For example, the use of signed binary arithmetic is highly effective for the implementation of fast adders, which in turn can be used in the implementation of pipelined IIR filters [2].

4. Design examples

4.1 G.726 and G.727 cores The hierarchical approach adopted can be illustrated by considering the design of an ADPCM encoder. This is shown in terms of a block diagram in Figure 2. An ADPCM decoder has a comparable structure. Figure 3 then shows a lower level description of the blocks required to implement the adaptive predictor and reconstructed signal calculator (within the dotted lines). It is immediately obvious that blocks such as FMULT are required several times in this block diagram(in this case with the same parameters). Other blocks such as LIMC, LIMD are also repeated throughout the designs (in this case with different parameters). Both blocks are available in the arithmetic library. The methods described have been applied to the design of various high performance ADPCM G.726 [4] and G.727 [5] CODECs. Four modules have been created, each of which fully support the four standards: G.726, G.726a, G.727, and G.727a. These offer on-line reconfigurability for different compression rates, PCM laws and ADPCM standards and have been fully tested using the ITU G.726 and G.727 test vectors.

4.1.1 Basic CODEC The basic CODEC has the simplest level of control. One encode or decode operation takes place in a single clock cycle. Two separate blocks are used, one for encoding and the other for decoding. The performance of a 0.6 µm Compass TLM CMOS standard cell design is summarised in table 2 with a layout in figure 4. This has an area of 2.7 x 2.1 mm^2.

4.1.2 Single Channel Duplex CODEC

The single channel duplex CODEC core provides both encoding and decoding based on a time-sharing mechanism. The encoding process has a latency of 3 clock cycles, whilst the decoding latency is 4 or 6 clock cycles depending on the core configuration. Table 3 gives the performance of this core for the same technology as above. From this, it can be seen that the critical path is greatly reduced, due to the increase in the level of pipelining to allow for the duplex operation. This is also apparent from the increased latency of the core, when compared to the basic CODEC.

4.1.3 Multi-Channel CODEC Core

For multiple channels, a single block has been developed which performs multi-channel encoding, decoding, or duplex coding. To cope with multiple channels the delay lines are implemented as RAM, thereby generating a highly efficient architecture. To calculate the gate count for each version, one must add 280 bits RAM/ channel to the values shown in table 4. At a 20 MHz clock rate, this core it is capable of providing up to 70 channels duplex coding, or 140 channels encoding or decoding.

	Encoder	Decoder
Gate Count	17k Gates	20k Gates
Critical path delay	120 ns	140 ns

Table 2. Basic CODEC performance.

.

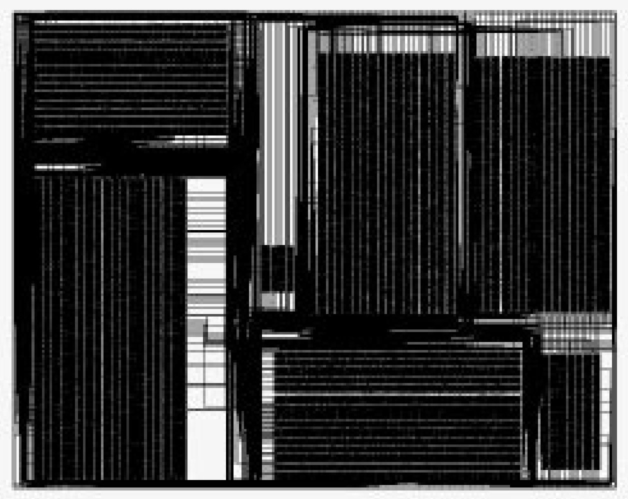

Figure 4. Basic encoder silicon layout

	Encoder/decoder
Gate count	16.7k Gates
Critical path delay	48 ns

Table 3. Performance of single channel duplex CODEC

	Multi-channel CODEC
Gate count	14.8k gates
Critical path delay	48 ns

Table 4. Performance of Multi-channel CODEC

4.2 Parameterised DCT core The second example is that of a generator which has been created to enable the rapid design of DCT cores. In this case, the focus has been on 8 x 8 DCT implementations in which wordlengths are parameterised. Input values are in the range 8-16 bits, coefficient values in the range 10-18 bits with an internal wordlength between 12 and 24 bits. The design shown (again for the same process) requires 56.5K gates, occupies an area of 4.48 x 4.29 mm^2 and has a critical path of 18ns. This provides a sampling rate of up to 55.5 Megasamples per second. Design times are similar to those above.

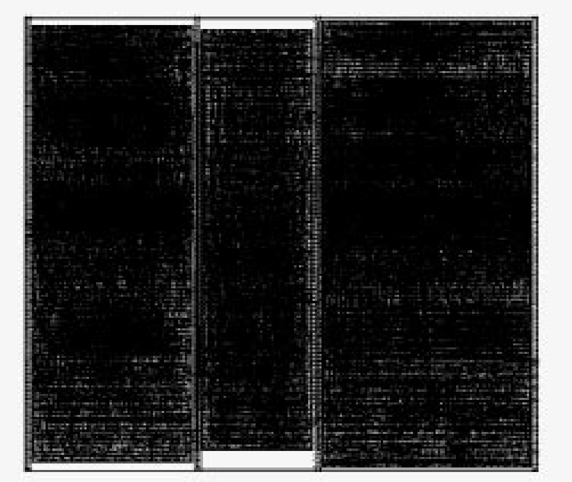

Figure 5. 8 x 8 point DCT core with 12 bit input data

5. Summary

An overview has been given of methods for the rapid design of advanced DSP ASICs and DSP ASIC cores based on the use of hierarchical VHDL libraries. This results in designs which are portable across many silicon foundries and comparable in area, performance and power consumption to ones based on more conventional methods. Design times are often reduced by several orders of magnitude. Recent re-designs in a 0.35 µm CMOS technology indicate that expected enhancements in performance are also achievable a simple example being a 300 MHz, 16 x 16 bit multiplier.

6. References

[1] J V McCanny and J G McWhirter "Some Systolic Array Developments in the UK", IEEE Computer, 1987, pp 51-63

[2] O C McNally, J V McCanny,R F Woods, "The Design of a highly Pipelined Second Order IIR filter chip" Intl. Jour High Speed Electronics and Systems" vol. 4, no 1,1993, pp 65-84

[3] C Hui, T J Ding, J V McCanny, R F Woods "A New FFT Architecture and Chip Design for Motion Compensation based on Phase Correlation" IEEE Trans. on Solid State Circuits, vol 31, no. 11, Nov. 1996 pp, 1751-1761

[4] ITU, Recommendation G.726, 40, 32, 24, 26 kbit/s ADPCM, CCITT, Geneva 1990.

[5] ITU, Recommendation G.727, 5-, 4-, 3- and 2-bits Sample Embedded ADPCM , CCITT, Geneva 1990.