LOW POWER ASYNCHRONOUS DSP FOR DIGITAL MOBILE PHONES

M. Lewis[1], L. Brackenbury[2]

Abstract

The CADRE digital signal processor (DSP) architecture is presented. This DSP is intended for use in digital mobile phones and, in this application, it is necessary to balance the requirement of high processing throughput with the demand of low power for extended battery lifetime. These requirements are addressed by a multi-level power reduction strategy, involving the use of a parallel asynchronous architecture, a configurable compressed instruction set, a large register file, the use of sign-magnitude arithmetic, and reduced support for interrupts.

1. Introduction

The market for mobile communications devices, particularly mobile phones, has become huge in recent years and is still growing rapidly. Associated with the growth of this market has been a vast drop in price for the phones themselves, with a myriad of different products from various manufacturers competing in the marketplace. The requirement for extended battery lifetime with reduced battery size makes this a key application for low-power VLSI design techniques.

Modern digital cellphones, conforming to the European GSM standard, execute complex control and signal processing functions, to perform filtering, error correction, speech compression and decompression algorithms (codecs), protocol management, and increasingly additional functions such as voice recognition or multimedia capability. This workload means that the digital components of these systems consume a significant proportion of the total system power. A typical basis for these digital components is a single chip, containing a microprocessor coupled by an on-chip bus to a DSP core. The microprocessor performs the control tasks, while the DSP is responsible for the intensive numerical processing. A study of the literature for one of these systems, produced by an industrial collaborator, showed that the DSP is responsible for approximately 65% of the total power consumption when engaged in a call using a half-rate speech codec. It can be expected that future generations of GSM chipsets will require even greater throughput from the DSP, to implement advanced low bit-rate codecs and to incorporate additional user features. This means that the total proportion of the power required by the DSP is likely to increase.

To tackle this problem a study is underway, as part of the EPSRC/MoD Powerpack project, investigating the design of an asynchronous digital signal processor to address the requirements for performance, power consumption and EMC arising from this application. This study has resulted in the CADRE DSP architecture (Configurable Asynchronous DSP for Reduced Energy) presented in this paper.

2. Sources of power consumption

Power dissipation in an on-chip processing system as described here can be broken down into two main areas. The first main area is the power cost associated with accesses to the program and data memories. This is made up of the power consumed within the RAM units themselves, and the power required to transmit the data across the large capacitance of the system buses. Memory accesses can form the largest component of power consumption in data-dominated applications [1],[2].

The second main area of power consumption comes from the energy dissipated in performing the actual operations on the data within the processor core. This is made up of the energy dissipated by transitions within the datapath associated with the data, and the control overhead required to perform the operations.

3. A new DSP for GSM chipsets

Our collaborator has suggested that the next generation of GSM chipsets will require more than 100MIPS throughput

1. mike.lewis@mic.ericsson.se; Ericsson Microelectronics AB, Isafjordsgatan 16, S-164 81 Kista, Sweden
2. linda@cs.man.ac.uk; AMULET Group, Department of Computer Science, Univ. of Manchester, Oxford Road, Manchester M13 9PL, UK

from the DSP. An initial target for throughput of 160MIPS has been chosen for the new design, which is intended to comfortably meet the requirements for this application. Should the achieved throughput exceed the requirements of a given situation, then the supply voltage could be reduced to give quadratic power reduction. However, if the supply voltage is fixed then the use of asynchronous design means that excess speed will be converted into power savings during the idle period at the end of the processing block. Asynchronous circuits inherently consume virtually no current when idle, due to the lack of a clock, and can go from idle to full activity instantaneously. Synchronous circuits use clock gating techniques to stop the clock; however, the idle time at the end of a processing block would not be sufficient to allow clock gating to be used. The properties of asynchronous design mean that the challenge can be thought of in terms of minimising the energy required for the given processing task.

DSPs are traditionally optimised for performing tight numerical processing kernels and are traditionally less good at control-oriented code. In the proposed application, the DSP will be working alongside a general purpose processor. The DSP can thus be operated as a coprocessor, performing tasks as directed by the microprocessor. The reduced control overhead greatly simplifies design of the DSP, thereby improving power efficiency.

4. Processor architecture

It has been shown that energy-efficient high performance circuits can be produced by exploiting parallelism [3]. This reduces the switching rate at each functional unit, with benefits both for power consumption and reduced electromagnetic interference. Silicon die area can be traded for increased speed, allowing simpler and more efficient circuits to be used or the supply voltage to be reduced. Silicon area is rapidly becoming less expensive; indeed, one of the challenges is to actually make effective use of the vast number of transistors available to the designer. This makes parallelism and replication very attractive, and for this reason a parallel structure with four independent functional units has been chosen. Analysis of key DSP algorithms showed that they can be readily parallelized, and multiple functional units allow algorithmic transformations to exploit correlation between successive data for reduced power consumption [4],[5]. The functional units need not be identical, meaning that different units can be substituted for a particular application. Asynchronous design makes this task easier, as the interfaces between blocks are defined independent of any global timing constraints. A block diagram of the architecture is shown in figure 1.

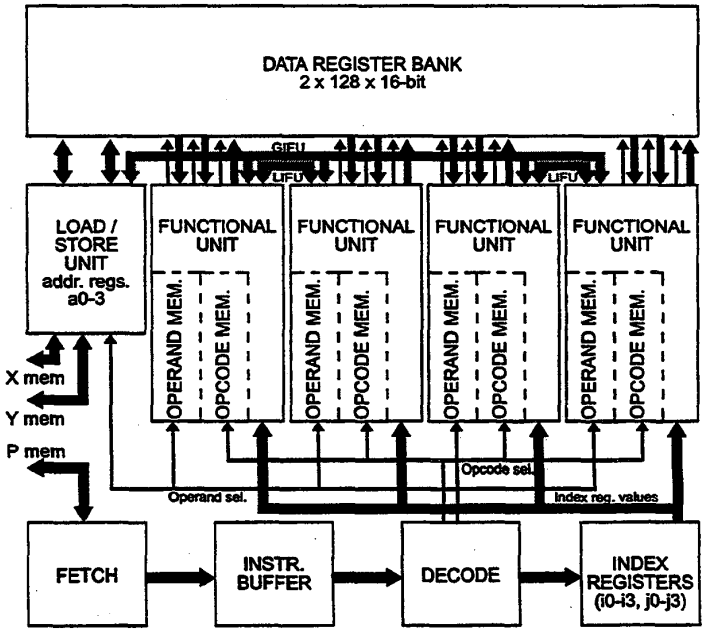

Figure 1 Block diagram of the CADRE DSP architecture

4.1. Reducing memory accesses

Having chosen a parallel architecture, a means of distributing instructions to the available resources is required. In contrast with general-purpose microprocessors, DSP activity can often be characterised by frequent repetition of a few fixed algorithms. This makes it possible to store parallel instruction encodings in advance, within configuration memories internal to the DSP. These configuration words can then be recalled with a single 32 bit compressed instruction, which allows a throughput of 160MIPS to be sustained from a system speed of only 40MIPS. A side-effect of the highly compressed instructions means that it is possible to execute complex DSP algorithms entirely from within an internal buffer of 32 instructions. The program memory is only accessed for the first pass through the loop, with subsequent iterations being supplied by the instruction buffer. This also maintains the loop counter, meaning that subsequent stages see an entirely flat instruction stream.

The configuration memories are located within the functional units, to minimise the capacitance of the associated wiring, and consists of two banks of 128 words. The first bank is the operand memory, which selects the sources and destinations of the data for each operation. The second bank is the opcode memory, which sets up the operation to be performed. The memory is partitioned in this way to maximise the reuse of configuration words. In addition, any component of the operation can be disabled from within the compressed instruction word, which also helps allow the reuse of the configuration words. An operation is defined by a particular operand address and a particular opcode address, used by all of the functional units.

A similar technique, where complex instructions are stored in a configuration memory, has been developed for a commercial DSP [6]. However, the authors believe that the design proposed here is significantly different, being more modular as the configuration memories are integral to the functional units, and more flexible as individual instruction components can be enabled and disabled, and few constraints are placed on the design of the functional units.

Having chosen a parallel structure, the next challenge is to supply data to each functional unit at a sufficient rate while keeping the power consumption to a minimum. The memory hierarchy approach works well for DSPs, as many algorithms display strong locality of reference or work on small blocks of data. For this reason, a large register file of 256 by 16 bit words was chosen, segmented into two banks labelled X and Y to match memory. The segmentation is algorithmically convenient in many cases, and also reduces the number of ports required for each bank.

The large register file allows for a high degree of data reuse, and a large explicit register file offers a significant advantage over a cache and fewer registers as is common in traditional DSP architectures. In the programmer's models of most traditional DSP architectures, operands are treated as residing within main memory and are accessed by indirect reference through address registers. These must be wide enough to address the entire possible data space of the processor, which is 24 bits in this design. After each operation, it is generally necessary to update these address registers to point to the next data item, which means that even if the data resides in the cache there is still a significant power consumption associated with these updates, and this power must be added to the power consumed by the cache lookup. The total power consumption from these factors is potentially large, as each functional unit can require up to two operands per operation.

In the new architecture, the address registers are used only for loading or storing data in bulk to and from the data register file; 32-bit ports to both X and Y memory allow up to 4 registers to be transferred simultaneously. Accesses to the data are then made indirectly by means of 7-bit index registers, which can be updated much more quickly and at much lower power cost than the wide address registers.

The combination of the large register file and the compressed instruction buffer can massively reduce the number of memory accesses: for example, it is possible to perform a 64-point complex FFT with only a single pass through both the program and data memory.

4.2. Reducing core power dissipation

Having tackled the power cost associated with memory transfers, the next area of attack is the power consumed within the processor core. It has been shown that sign-magnitude number representation can offer reduced switching activity compared to two's complement numbering when data are correlated. This is due to the large number of redundant ones required to represent a small negative value in two's complement form. However, sign-magnitude representation requires more complex arithmetic circuitry, particularly when two numbers of differing sign must be added. To investigate this trade-off, a study was performed based around simulated DSP operations on detailed models of DSP datapaths for both sign-magnitude and two's complement numbering. This found a reduction in switching activity of 10-55% when using sign-magnitude representation. The study did not take into account transitions on system buses or memory accesses, which should make the real reduction even greater. Also, the extra complexity for sign-magnitude arithmetic is in minimum-geometry sections of the datapath

and should contribute little to the total power consumption. For these reasons, sign-magnitude representation of data was chosen for the design of the functional units.

Asynchronous design techniques were chosen for the processor, based on the principle of micropipelines [7] where each processing stage negotiates the passing of data to its neighbours by means of request and acknowledge handshake signals. Architecturally, there is no overriding reason why synchronous design techniques could not have been chosen, but asynchronous design has a number of compelling advantages. Firstly, the lack of a clock distribution network eliminates the associated power consumption, and means that clock gating is unnecessary as mentioned earlier. Secondly, asynchronous designs emit very much less electromagnetic radiation than synchronous designs, which is very important for wireless devices. Finally, asynchronous design gives a modular design style, which allows arbitrarily complex designs to be produced by means of well-defined interfaces between blocks, without worrying about the problems of global clock distribution.

5. Testing and evaluation

CADRE has been completed to the schematic design stage, and consists of approximately 750,000 transistors in a 0.35µm 3 metal layer CMOS process. Testing has been performed by simulation of netlists taken from the schematics, using Synopsis' *Powermill* simulator using typical silicon parameters at 3.3v. Powermill is claimed to be close to SPICE in its accuracy, at a fraction of the computational load. Power consumption probes were assigned in a hierarchical manner, to provide a breakdown of the power consumed by the various segments of the design. Also, the simulator was set up to monitor various architectural features such as the number of register and memory accesses and the average occupancy of the functional units.

The tests were performed with three DSP algorithms, to establish the performance and power consumption of the processor: a 20 point FIR filter, a 64-point complex FFT and the preprocessing and linear predictive coding (LPC) analysis section of the GSM full-rate speech compression algorithm. The FIR filter and FFT each processed 256 data samples, while the LPC analysis algorithm was performed on a GSM data frame of 160 samples. To evaluate the impact of data characteristics on power consumption, the FIR filter and FFT algorithms were run separately on random data and speech data (extracted from the ETSI speech test sequence used for testing GSM codecs). The LPC analysis algorithm was run only on speech data.

In a complete system, the memory power consumption may be a significant proportion of the total power consumption. In the simulations, the memories were implemented using C behavioural models. To estimate memory power consumption, the models were designed to report power consumption to the simulator during each read or write access, so as to consume a fixed amount of energy for each operation. The energy per operation was estimated at 0.67nJ, which was based on measurements of power consumed by the 8 kilobyte RAM block of the AMULET3i asynchronous embedded island [8].

6. Test results

6.1. Instruction execution performance

Operating speed results for the three algorithms are shown in Table 1. This shows the rate of issue of parallel instructions, the operation rate within the functional units, and the average proportion of the functional units which are occupied for each parallel instruction.

Test	Instruction rate	Arithmetic operation rate	Occupancy
FIR filter	43MHz	163MOPS	95%
FFT	38MHz	141MOPS	93%
LPC analysis	34MHz	117MOPS	86%

Table 1. Parallel instruction issue and operation rates

The instruction rate is the measured rate of dispatch of parallel instructions to the functional units. This value depends on how many control / setup instructions had to be inserted between parallel instructions, and also on how quickly register reads and arithmetic operations completed. The arithmetic operation rate is the measured rate of arithmetic operations within the functional units, which depends on the instruction rate and the occupancy (how frequently each functional unit is used in parallel instructions).

It can be seen that the operation rate for the FIR filter exceeds the 160 MOPS target: the FIR filter kernel is extremely efficient, without any setup code required once the kernel is underway. The FFT algorithm is somewhat less efficient, requiring changes to the index registers between successive passes of the FFT kernel. Since the speed of arithmetic operations is not data dependent, the same operation rates were observed for both speech and random data. The GSM LPC analysis program is the least efficient, as the test involves a number of separate algorithms applied sequentially which require setup instruction between each pass. Also, some of these

algorithms cannot be partitioned easily across the functional units. This is evident in the reduced utilisation figure.

6.2. Power consumption results

Average power consumption for each of the algorithms is shown in Table 2. The run time over which power consumption is measured extends from the moment that the reset signal is removed to the time that the tests are completed. The distribution of the energy consumed to the various different components of the architecture is shown in figure 2.

	FIR		FFT		GSM
	rand.	speech	rand.	speech	
Power (mW)	668	584	676	660	406
Run time (μs)	38.9	38.5	32.7	32.5	16.1
Arithmetic ops.	5888	5888	4100	4100	1288

Table 2. Power consumption, run times and operation counts

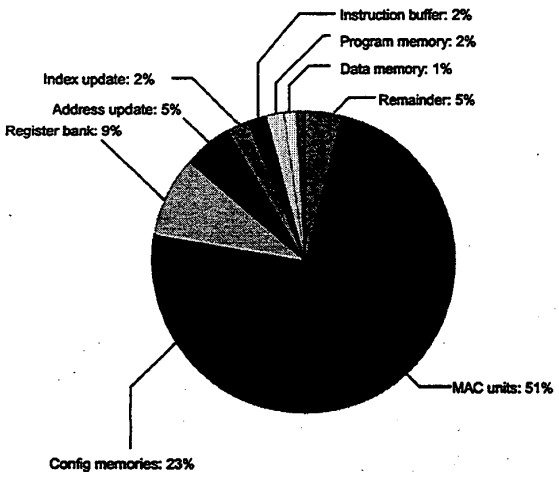

Figure 2 Distribution of energy per operation

It must be emphasised that the results are based on simulated schematics, which do not include extracted parasitic capacitances of interconnections. However, CADRE is designed specifically to minimize the distance over which data must travel and therefore it is expected that the inclusion of post-layout capacitances will not seriously affect the performance, nor unduly affect the power consumption.

6.3. Evaluation of architectural features

6.3.1. Register bank

The register bank is accessed between 6 and 21 times more frequently than the memory, and the register bank consumes on average 3 times less energy per access than the memory system despite being highly multi-ported.

6.3.2. Index registers

Between 8 and 22 times more updates are performed using the small index generation units rather than the address generation units. Updates to the index registers require an order of magnitude less energy than updates to the address registers, leading to a significant reduction in power consumption.

6.3.3. Instruction buffer

The measured energy per instruction passing through the buffer is between 32% and 45% of the estimated energy required to fetch a word from program memory. The measured ratio of instruction issued by the buffer to those fetched from memory varies from between 2.7 and 22, depending on how efficiently a given algorithm makes use of the DO construct.

6.3.4. Effect of sign-magnitude arithmetic

An indication of how much benefit is obtained by the use of sign-magnitude numbering can be gained by comparing the results for uncorrelated (random) data with those for correlated (speech) data. The energy consumption figures for the FIR filter algorithm show a total reduction in energy per operation of 13% when processing speech data rather than full-range random data. The figures for the FFT algorithm performed on comparably scaled speech data show a reduction of only 1%: the FFT algorithm is such that adjacent data points tend not to be processed sequentially, reducing the amount of correlation that can be exploited.

6.3.5. Effect of asynchronous design

The benefits of using asynchronous design stem primarily from the ability to automatically shut down during idle periods: the benefits given by this will depend on the constraints of a given practical system, and cannot be evaluated by 'flat-out' performance testing as presented here.

6.4. Comparison with other designs

Table 3 compares performance of CADRE with two commercial DSP architectures, which were marketed as

'low-power' at the start of the CADRE project. Manufacturers' figures are generally presented at the best operating conditions; therefore the figures presented for CADRE are those estimated for 1.2v operation. The chosen metric for comparison is energy-delay product: energy per operation can be reduced at the expense of delay by varying the supply voltage and reducing gate drive strengths, and vice-versa. Energy-delay product gives a measure of quality for the underlying design and technology.

Design	Tech.	Power	Speed (MOPS)	Energy-delay (nJ.ns)
Product A	0.6µm	96mW	40	60
Product B	0.6µm	83mW	80	13
CADRE	0.35µm	29mW	55	10

Table 3. Comparison of CADRE to commercial designs

The figures show CADRE to have a significantly better energy-delay product than Product A, and a slightly better energy-delay product than Product B. However, CADRE was designed using a more advanced process technology, and the figures for CADRE do not include the effects of wire parasitics. These results are therefore somewhat disappointing: given the benefits being obtained by the architectural features, a clearer advantage was expected.

The main culprit in the reduced performance may be seen in the breakdown of energy consumption in figure 2: 51% of the total power goes into the multiply-accumulate units. The design of CADRE was performed over a very limited period of time, which meant that little time could be spent selecting and optimizing particular circuits. A re-design of the multiplier and adder circuits has produced circuits with energies reduced by factors of approximately 3 and 6 respectively: other components of the design may also benefit to a similar extent. Furthermore, the multiply-accumulate operation was unpipelined (for simplicity) which has caused a greatly reduced maximum operating speed: a pipelined speed of approximately 400 MOPS seems readily attainable at 3.3v. Estimates for performance with pipelined operation suggest that an energy-delay product of 3.3nJ.ns could be attained.

7. Conclusions

The CADRE DSP architecture has been presented. This uses a variety of architectural techniques aimed at reducing power consumption for the next generation of mobile phone applications. The results show CADRE to easily meet the speed requirements that were set out. However, the energy-delay metric for the design was less good than expected.

The outcome emphasises the importance of considering all levels of the circuit in low-power design: time constraints prevented circuit optimization, which led to excessive power consumption in the arithmetic elements and possibly elsewhere. This obscured the benefit obtained by the architectural features. With optimization, including a pipelined multiply-accumulate, the techniques used in CADRE appear set to offer new levels of performance and energy efficiency. Many of the design decisions in the CADRE architecture were intended to combat the problems of deep sub-micron processes; notably wire parasitic effects. The benefits of the CADRE architecture should become clear when transferred to 0.18µm or smaller technology, and this work is currently in progress.

8. Acknowledgements

This work formed part of the EPSRC/MoD Powerpack project, grant number GR/L27930. The authors wish to express their gratitude for this support.

9. References

[1] F. Catthoor, "Energy-Delay Efficient Data Storage and Transfer Architectures and Methodologies: Current Solutions and Remaining Problems", Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology", vol. 21 no. 2, pp 258-265, 1999

[2] K. Danckaert, K. Masselos, F. Catthoor, H.J. DeMan, C. Goutis, "Strategy for Power-Efficient Design of Parallel Systems", IEEE Transactions on VLSI Systems, vol.7 no. 2, pp219-231, 1999

[3] A.P. Chandrakasan, R.W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits", Proc. IEEE vol. 83 no. 4, April 1995

[4] T. Arslan, A.T. Erdogan, D.H. Horrocks, "Low Power Design for DSP: Methodologies and Techniques", Microelectronics Journal, vol. 27, no. 8, pp 731-744, Nov. 1996

[5] A.T. Erdogan, T. Arslan, "Low Power Multiplication Scheme for FIR Filter Implementation on Single Multiplier CMOS DSP Processors", Electronics Letters, vol. 32, no. 21, pp 1959-1960, 1996

[6] P. Kievits, E. Lambers, C. Moerman, R. Woudsma, "R.E.A.L. DSP Technology for Telecom Baseband Processing", Proc. 9th Intl. Conf. On Signal Processing Applications and Technology, Miller Freeman Inc., 1998

[7] I.E. Sutherland, "Micropipelines", Communications of the ACM, vol. 32, no. 6, pp 720-738, June 1989

[8] J.D. Garside et. al., "AMULET3i - An Asynchronous System-on-Chip", Proc. *International Symposium on Advanced Research in Asynchronous Circuits and Systems*, April 2000, pp. 162-175