THE APPLICATION OF A NOVEL ADAPTIVE DYNAMIC VOLTAGE SCALING SCHEME TO SOFTWARE DEFINED RADIO

Craig Dolwin (Toshiba Research Europe Ltd, Bristol, UK, craig.dolwin@toshiba-trel.com)

ABSTRACT

This paper presents a novel technique for using Adaptive Dynamic Voltage Scaling (A-DVS) in Software Defined Radio (SDR). It is well known that high power consumption is a major stumbling block to the introduction of SDR into the commercial handset market even so it is highly desirable to use a generic System on Chip (SoC) in a wireless platform so that it can be reconfigured either when the handset is manufactured or at run-time. This approach addresses the ever demanding issues of increased Radio Access Technology (RAT) complexity and the associated high engineering costs. By using A-DVS in a generic SoC, you are able to dynamically and automatically optimise the resources to match each new configuration and therefore reduce the power consumption.

1. INTRODUCTION

One approach to reducing power consumption, while still achieving the high level of flexibility required in SDR, is to use multiple voltage/frequency islands in a SoC [1]. This structure is known as a Globally Asynchronous Locally Synchronous (GALS) architecture. In a SDR platform each island would contain a programmable processing element which could be optimised for a specific class of signal processing (e.g. channel codec, source codec, fft etc) [2]. The configuration process involves the setting of parameters and linking of modules through a communication fabric. While in principle this seems fairly simple in practise special care has to be taken to schedule the multiple processing elements so all hard real-time deadlines are met. In such a complicated operating environment the execution time of a task might be altered by other processors contending for a shared resource e.g. a data bus. The precise interaction between resources and the impact on a tasks execution time is very hard to predict for a given configuration so by using an adaptive scheme as proposed in this paper the unpredictable timing delays can be accommodated while still achieving the desired timing deadline and operating at the lowest possible supply voltage.

The proposed A-DVS scheme is based on calculating a voltage-frequency profile for the duration of an operation executing. This profile is made up of a set of voltages-frequencies which increase as the operation executes. If the operation completes before its Worst Case Cycle Count (wccc) the average operating voltage will be reduced but by choosing the profile carefully the operation will still complete by the required deadline if it uses the worst case number of cycles.

The paper illustrates two methods of implementing A-DVS and shows how power can be saved in an SDR application and reports on the implementation complexity using fixed point C on a Texas Instruments DSP.

2. ADAPTIVE DYNAMIC VOLTAGE SCALING

The designer of a CMOS VLSI has a number of techniques that will help reduce the power consumption in a circuit [3] . One such technique is the optimum setting of the supply voltage. The average dynamic power consumption (Pavg) of a node in a CMOS circuit is defined by Eqn 1 :

$$P_{avg} = \alpha_{0\to1} C_l V_{dd}^2 f_{clk} \qquad (1)$$

Where $\alpha_{0\to1}$ is the transition rate on the node, C_l is the load capacitance, f_{clk} is the clock frequency and V_{dd} the supply voltage. From this equation it can see that a quadratic improvement in power consumption can be achieved with a reduction in supply voltage. Unfortunately the supply voltage also determines the maximum operating frequency. The engineer must therefore select the voltage to match the maximum operating frequency. In many simple data processing functions this is easily quantified but with the introduction of more complicated algorithms with variable cycle counts, multi – mode wireless platforms and Software Defined Radio (SDR) this calculation becomes problematic. Eqn 2 states that for each cycle of execution there is an optimum clock period [4][5].

$$\tau(n) = T_{deadline} \frac{\sqrt[3]{1-cdf(n)}}{\sum_{i=wccc}^{i=0} \sqrt[3]{1-cdf(i)}} = T_{\max} \sqrt[3]{1-cdf(n)} \qquad (2)$$

Where τ is the clock period for cycle n, wccc is the worst case cycle count and cdf(n) is the probability that the task will use no more than n cycles to complete.

To meet the timing deadline the period of the clock supplied to the function must be equal to or less than the value of τ.

The model used to represent the baseband processing of an SDR transceiver is a set of tasks each of which are periodically repeated on a frame by frame basis and each

have a start time and a deadline ($T_{deadline}$) by which they must complete. Each task has variable complexity i.e. it takes a variable number of cycles to complete. The statistics of this variable cycle count are not known at design time, all that is known is its wccc. The variability in cycle count can be due to conditional branches in the execution of the task, dynamically variable execution parameters (e.g. Adaptive Multi-Rate Speech Codec) and resource contention (e.g. communication across a shared data bus). To maintain the hard real-time deadlines for a specific task in a reconfigurable system the wccc must be calculated for the task in all of the possible configurations. This calculation is not trivial and will be based on the designer's knowledge of the longest path through the code as well as its interaction with external resources (e.g. reading and writing to global memory). Any platform claiming to support real-time operation must be able to support deterministic communications between resources.

In Figure 1 the relationship between a frame period, a task execution period and the dynamic variable supply voltage is shown. The supply voltage (and therefore also the clock frequency) starts low and increases until the timing deadline is met. The task itself will complete at some point in this period depending on the specific characteristics of this execution.

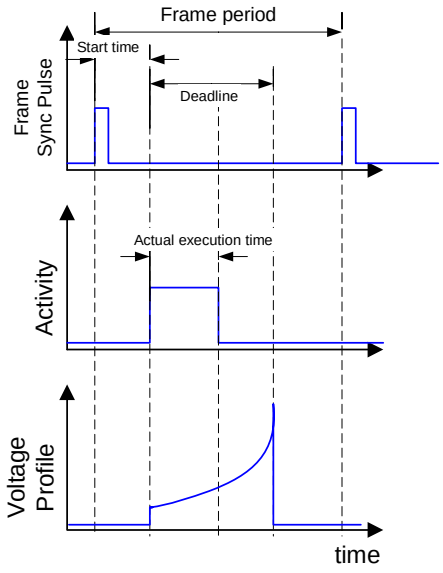

Figure 1 Relationship between frame period, task activity and voltage profile

To calculate the voltage profile and therefore the clock period profile the Cumulative Distribution Function (cdf) must be obtained to do this the past history of cycle counts for this task is used. This means the initial voltage profile will be inaccurate but this only means the first few executions of the task are executed inefficiently but the task will still complete on time. As the statistics become more reliable the power saving should improve. To guarantee that the timing deadline is always met the area under the clock period vs clock cycle profile must always be less than $T_{deadline}$, Eqn 3. If the actual clock cycle period actually matches Eqn 2 then it can be easily shown that the deadline will always be met.

$$T_{deadline} \geq \sum_{n=0}^{n=wccc} \tau(n) \qquad (3)$$

Two methods were investigated for calculating the statistics. The first method [6] (The Three Phase Method) assumed the distribution could be approximated using a simple three phase piecewise linear distribution. After each execution of a task the updated cycle count history is used to calculate the values for C1 and C2. Where C1 and C2 are thresholds corresponding to the assumed simplified distribution, see Figure 2.

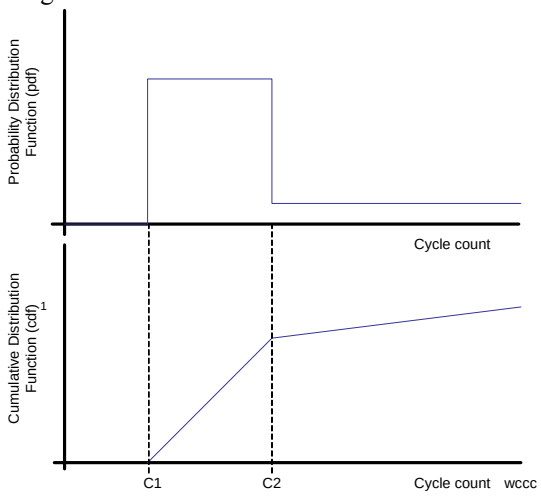

Figure 2 Three Phase Method for A-DVS calculation

Taking C1, C2 and Eqn 2 a three phase voltage profile is calculated. The Three Phase method simplifies the calculation of the profile but assumes the supply voltage can be set to any value within a range. In practise a silicon technology may only be characterised for a limited set of operating voltages and clock frequencies. So for the second method the operating voltage and therefore clock period was constrained to a limited set of values (The Voltage Constrained Method) [7]. This method uses a set of counters, also known as bins, to build up a pdf of the cycle count. The counters are then used to calculate the cdf and then the profile of the clock period using Eqn 2. Then using the available set of clock periods, starting with the longest clock period the algorithm calculates, by indexing into the clock period profile array, the minimum amount of time the resource can stay in each phase, where a phase corresponds to a fixed clock period and its associated voltage.

3. RESULTS

Three Phase Method

To test the Three Phase Method a transaction based SystemC model for an advanced SDR SoC as illustrated in Figure 3 was developed. The SoC is made up of multiple processor resources and a control processor all of which are linked by two busses, one for control information and a second for transferring data between processing resources. Each processing resource has its own supply voltage and clock frequency. The control voltage and clock frequency are set by the control processor. To represent a sophisticated SDR platform without having to implement the detailed processing the functionality in each processor was simplified to a task that simply reads a block of data from global memory and then writes it back after a variable number of cycles. The data-bus is shared between processors and so an arbitration unit is required to determine who has access to the bus. In this simulation the process of transferring data from a resource to global memory is similar to the operation of a Dynamic Memory Access (DMA) module. Each resource requests a data transfer from the DMA module and the arbitration unit determines when the resource gets access to the data bus based on the priority of the request. The random nature of this contention will also add a random delay to the completion of each task and this must be dealt with by the A-DVS scheme.

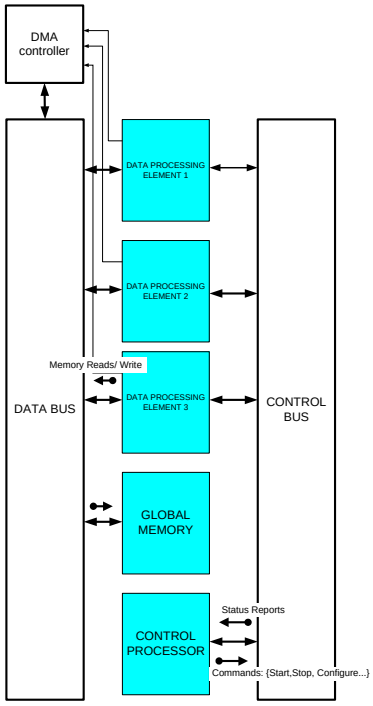

Figure 3 SystemC model for GALS SoC

The simulation was run with an extra task being added every 20ms until a total of 20 tasks were executing on 20 resources. As each task was added, the contention between the resources when accessing the data bus also increased. This meant the A-DVS scheme had to adapt and modify the voltage-frequency profile appropriately. The execution time of a single task with all the other 19 tasks executing was monitored with and without A-DVS and from Figure 5 it can be seen that the A-DVS scheme shifts the mean completion time from a value close to 40us towards the deadline of 80us. This shows that using A-DVS the circuit will on average use the resource more efficiently i.e. it is not being left idle for as long as the fixed voltage scheme.

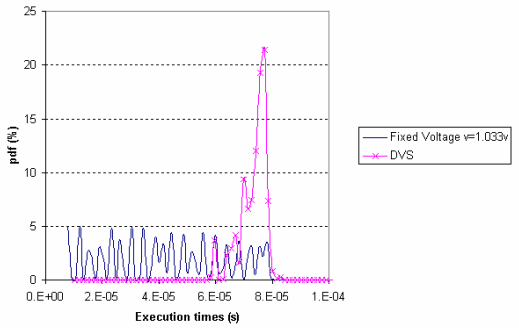

Figure 5 Task execution times with and without DVS

Voltage Constrained Method

The voltage constrained method was initially simulated using MATLAB and a function was created to simulate the cycle count of the task. The different types of tasks are defined in Table 1

TABLE 1
FUNCTION TYPES

Name	Description	WCCC	$T_{deadline}$ (ms)
Fixed	Cycle count fixed at 16000 cycles	32000	32
Flat	A uniform distribution of cycle counts between 0 and 32000	32000	32
Binomial	A binomial distribution of cycle counts with a mean of 16000 and a standard deviation of 2000	32000	32
AMR	The measured cycle counts for an AMR encoder. The average cycle count was 54000.	1.5e6	10

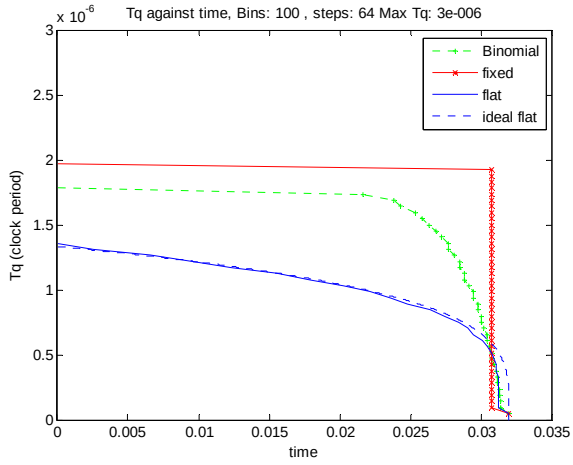

Figure 6 Clock period profiles for different function types

Figure 6 shows how the clock period varies over the duration of the task for each task type. These profiles can be explained as follows:

The *fixed* profile stays at a constant low frequency until it is close to the deadline and then immediately switches to the highest clock frequency. This is what is expected because the past history has told us the function is very unlikely to continue executing beyond this point but if it does the profile still ensures the deadline will be met by executing at the maximum frequency. Should the cycle count go to the wccc the power dissipation will increase dramatically but this should only happen vary rarely.

The *binomial* profile stays at a low frequency and then begins to gradually increase. The higher start frequency compared to the fixed profile and the gradual increase in frequency is due to the level of uncertainty about predicting the future cycle count.

The *flat* profile corresponds to the greatest uncertainty about what the future cycle count values will be and therefore starts at a higher frequency and steadily increases with time. This uncertainty in cycle count will mean the average power consumption will be higher than the binomial distribution despite having the same mean cycle count.

The MATLAB simulation was used to determine the optimum setting for the program parameters when applied to the AMR encoder. The parameters are summarised in Table 2.

TABLE 2
A-DVS parameters

Parameter	**Optimum Value**
No of bins when calculating the pdf	20 bins
No of Voltage values	> 8 (see Figure 7)
Frequency of Profile Calculation	10 speech frames
History	4 x 10 speech frames

Figure 7 presents how much power is saved using the A-DVS when normalized to a fixed voltage scheme. In the fixed voltage scheme the clock frequency and therefore the voltage are calculated based purely on the task always executing wccc cycles and the power is then calculated assuming the processor switches off when the task completes. The saving in power can be seen to increase with the number of clock frequencies supported but eventually reaches a floor. A large change in power saving is seen when switching from an ideal silicon model (Silicon 1) to a practical model (Silicon 2) as defined in Table 3.

TABLE 3
Platform Types

Name	Max Freq	Min Freq	Max V	Min V
Silicon 1	720MHz	90MHz	3.2	0.4
Silicon 2	382MHz	99.5Mhz	1.3	0.85

The voltage profile calculation was implemented in fixed point C on the Texas Instruments C62 DSP. The code was profiled using an optimum setting for the AMR encoder and a cycle count overhead was established as shown in Table 4. Given that the AMR encoder typically takes 54000 cycles to complete and the profile calculation only needs to be done every 10 frames it can be can seen that the A-DVS overhead is quite small i.e. less than 1%.

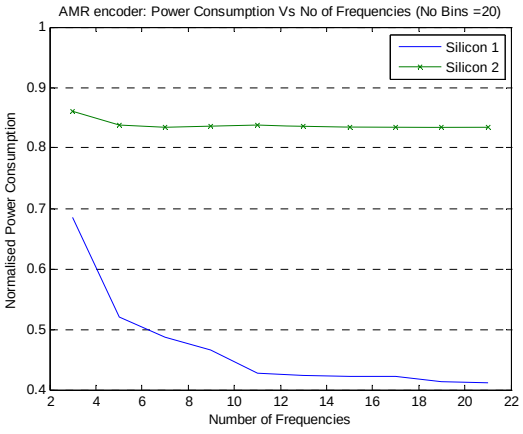

Figure 7 Normalised power consumption trends

TABLE 4
Algorithm Complexity Results

Parameter	Value	Floating Point Cycles	Fixed Point Cycles
Frame Size	10	4500	5280
Number of Bins	10		
Number of Frequencies	8		
Frame History	4		
Function Type	AMR encoder		

4. CONCLUSION

This paper briefly outlines two methods for implementing an A-DVS scheme. It has been shown that this approach can potentially offer significant power savings in reconfigurable hard real-time applications, such as SDR. It has also showed that despite the apparent complexity of the voltage profile calculation it can be simplified in a practical application so that the overhead is not significant when compared to complex signal processing operations such as the AMR speech codec.

The current limitations of this approach are down to the current availability of silicon with a sufficiently large range of operating voltage and clock frequency as well as the ability to switch between a large numbers of operating voltages.

ACKNOWLEDGMENT

I would like to thank Hatem Yassine for his valuable contribution to this work.

REFERENCES

[1] K. Niyogi and D. Marculescu, "Speed and Voltage Selection for GALS Systems Based on Voltage/Frequency Islands," in Proceedings of the 2005 conference on Asia South Pacific design automation, Shanghai, 2005.

[2] B. Heyne, J. Gotze, and M. Bucker, "Implementation of a Cordic Based FFT on a Reconfigurable Hardware Accelerator," in Karlsruhe Worksop on Software Radio Karlsruhe 2004.

[3] A. P. Chanrakasan and R. W. Brodersen, "Minimizing power consumption in Digital CMOS circuits," Proceedings of the IEEE, vol. 83, April 1995.

[4] F. Gruian, "Hard Real-Time Scheduling for Low-Energy Using Stochastic Data and DVS Processors," in International Symposium on Low Power Electronics and Design 2001 Huntington Beach (CA), US, 2001.

[5] J. R. Lorch and A. J. Smith, "PACE: A new approach to dynamic voltage scaling," IEEE Transactions on Computers, july 2004.

[6] C. A. Dolwin, "Application of dynamic voltage scaling to a reconfigurable wireless terminal," Workshop on Software Radio, Karlsruhe, 2004.

[7] C. A. Dolwin, "Evaluation of an adaptive dynamic voltage scaling scheme for hard real-time applications," Journal of Low Power Electronics, vol. 3, pp. 1-5, 2007.