Dynamic Voltage and Frequency Scaling in an Embedded Microcontroller SoC

Spring 2006 – ECE 6770

Nathaniel Gaskin, Amlan Ghosh, Spencer Kellis

I. Introduction

Given energy as the limiting factor in a low-power battery-driven embedded microcontroller (MCU), energy-efficient operation is critical. Low-energy microcontrollers benefit from comparable performance and longer lifetimes when compared to microcontroller cores without energy saving schemes. Dynamic voltage and frequency scaling (DVFS) has been proposed in [1] as a novel technique to provide optimal performance given application requirements in a microcontroller. This project involved the design of a similar structure which will be integrated onto the Wireless Integrated Microsystems (WIMS) MCU described in [2], which is targeted for embedded sensor applications.

The WIMS MCU employs a three-stage pipeline with 32 kB of on-chip SRAM and a 512-byte low-power loop cache. The MCU is driven by an on-chip MEMS-LC clock generator which can scale frequency in just over one cycle of MCU operation. With V_{DD} at 1.8V, the WIMS MCU runs at 100MHz.

II. Previous Work

Weiser, *et al* [3] introduced frequency scaling as a means of reducing CPU energy as early as 1994. Their work focused on scheduling algorithms, but suggested the need for both voltage and frequency scaling.

Pouwelse, *et al* in [4] implemented voltage scaling on an embedded StrongARM 1100 processor running a customized Linux kernel which dynamically controlled the voltage and frequency based on system workload. They showed an energy savings of 80% when the supply voltage was reduced from 2.0V to 0.8V and frequency was reduced from 251 MHz to 59 MHz.

Burd, *et al* proposed a technique in [1] which allows the operating system of a microcontroller to control the operating frequency. A hardware structure deals with the issue of mapping a frequency to an operating voltage. Feedback loops allow for error control. Since the operating system is the only system component with a bird's-eye view of the workload, the operating system was given control over scaling frequency. Applications can communicate their needs to the operating system.

This work differs and extends previous work by implementing independent control over frequency and voltage, for the highest scaling resolution possible. The operating system will analyze the needs of the system over time and provide fine-grained control over voltage and frequency independently to meet deadlines while saving the most energy possible. Two of the existing memory-mapped register in the WIMS MCU will be dedicated to the DVFS system.

III. Block-level Description

The WIMS MCU does not utilize an operating system in the traditional sense of the term; rather, a boot ROM loads application code and transfers control to that code, which runs until exit. This work assumes an oracle knows the correct operating frequency and writes that frequency to the memory-mapped register. The control circuitry in the DVFS system maps a new voltage and transitions to the required frequency.

IV. General Design Information

The WIMS MCU fits into 3.2mm x 3.5mm. The DVFS control logic will fit with an ADC into a region 1.2mm^2 as shown in Fig 2. Power consumption should be minimal compared with the memories and core of the MCU. [1] estimates that CMOS scales at approximately 100 V/us, and the MCU utilized by this project should scale comparably.

Figure 1Block diagram of the DVFS system

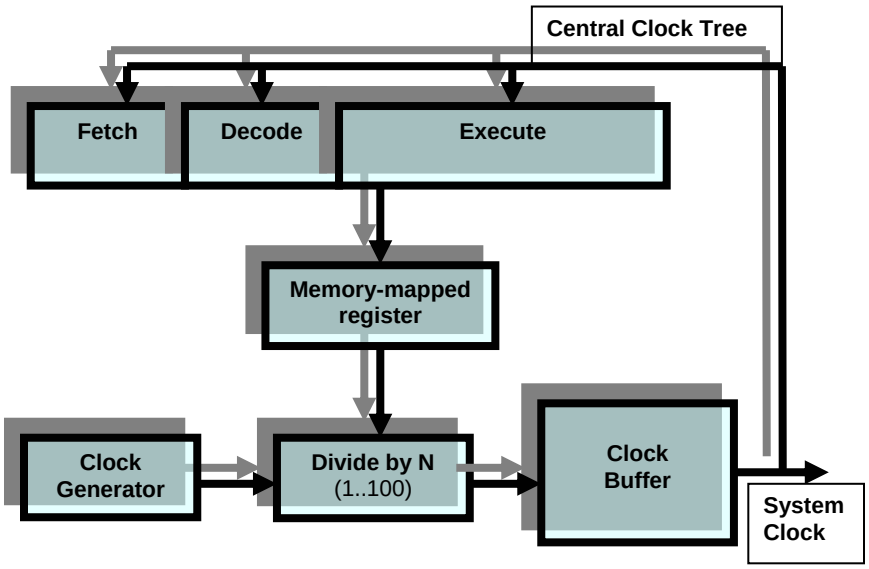
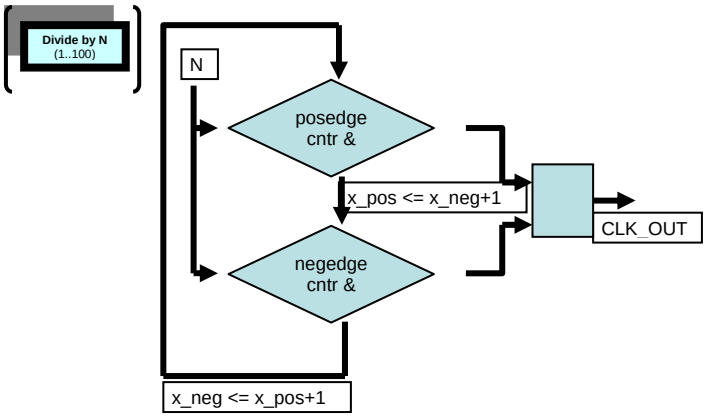


Figure 1 shows the basic block diagram of the dynamic frequency scaling circuitry. This digital synchronous frequency divider has been implemented to do the dynamic frequency scaling which can divide the clock by anywhere from one-half of its frequency to as aggressive as a hundredth of its frequency. This wide range of division provides flexibility in reducing the supply voltage of the core and hence reducing leakage power during the standby mode. This design style has been chosen over standard clock division by PLL to avoid the transition time and power consumption of a PLL, which have been known to require thousands of cycles to change frequencies. Moreover, the power consumption overhead to integrate a PLL with the MCU may not achieve the desired goal of reducing power.

Figure 2 Detailed diagram of the Divide-by-N control logic

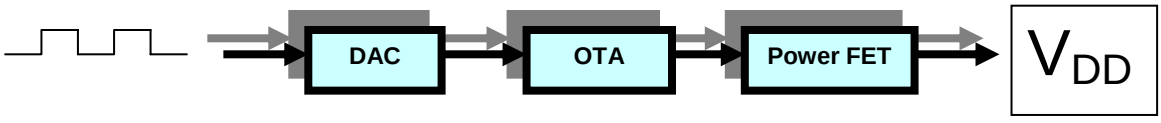


As seen in Figure 2, both positive edge and negative edge triggered latches are introduced in the design to generate the 50% duty cycle for odd number of divisions. One 64 driving capability clock buffer is added at the output to drive the system clock. The whole circuit works synchronously such that there will be no phase difference during the transition.

Our initial designed used a DC-DC converter to regulate voltage. DC-DC converters have shown higher efficiency than linear regulators. Our hope is that the simplicity of our implementation, with the required control or feedback circuitry of the DC-DC converter, will be significant enough to justify the design.

As in the case of the frequency scaling, a memory mapped register will be available, whose value will serve as the input to a linear voltage regulator module. The 8-bit value in the memory mapped register will drive a digital to analog converter, providing current to an amplifier. The output of the amplifier will drive a power MOS, allowing a regulated voltage to be applied to the chip as shown in Figure 3. Initial measurements on the WIMS microcontroller state that at least 14.3 mA of current will be needed to drive the core of the chip. Spice simulations are being performed to validate our design.

Figure 3Block diagram of the Voltage-Scaling Frequency control logic



A current-summing DAC has been designed using current mirror structures. The current mirror corresponding to each bit will be connected to either desired current level or ground. The size of the current-mirrored transistors determines the amount of current flowing through a particular branch; bit lines are connected according to their significance to a correspondingly sized mirror.

The binary weighted current summing DAC is show in Fig. 3, with its corresponding layout in Fig. 3. The DAC may be enhanced by using differential switched instead of pass transistors. The layout area is 38 um x 18 um.

Figure 3 DAC Schematic

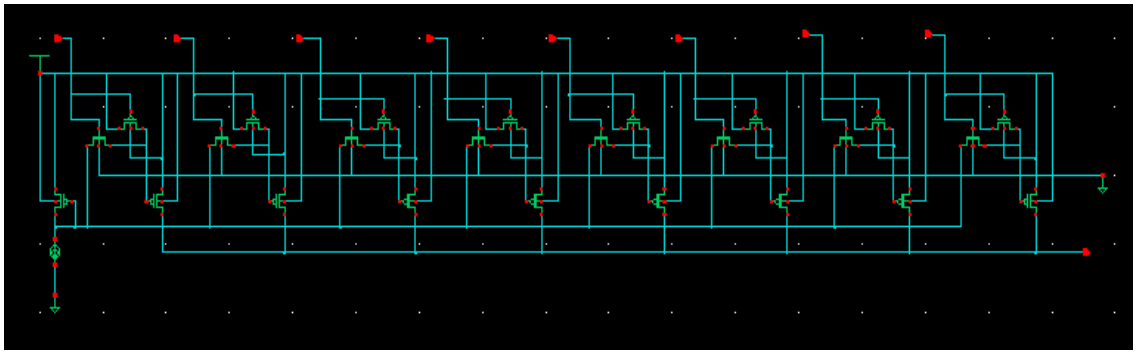
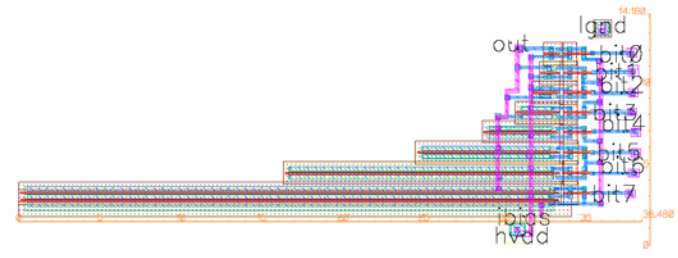


Figure 4 DAC Layout



A schematic diagram of the operational transconductance amplifier is shown in Fig. 5. A low-power low-noise current mirror operational transconductance amplifier has been used. Proper transistor sizing has been carefully determined to keep low noise at this current level. The bias current I_{bias} is controlling the operating region of all the transistor of the circuit which equally branches between M_1 and M_2 to generate half of the bias current flowing through all the devices. The inversion coefficient (IC) for each transistor may be calculated as the ratio of the drain current to the moderate inversion characteristics current I_s. A device having IC>10 operates in strong inversion whereas IC<0.1 shows a characteristics of sub-threshold region. Given this drain current, W/L ratios have been determined considering required transconductance for the devices and hence linearity of operating across the operating range, however, not violating stability of the circuit crossing the phase margin limit .

The layout of the OTA is seen in Fig 6. The IBM 130nm process rules were followed to obtain an area of 34 um x 14 um.

Figure 5 OTA Schematic

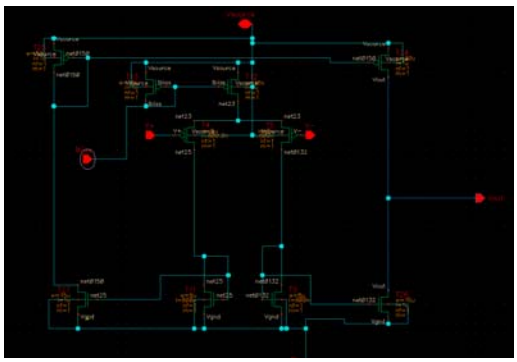
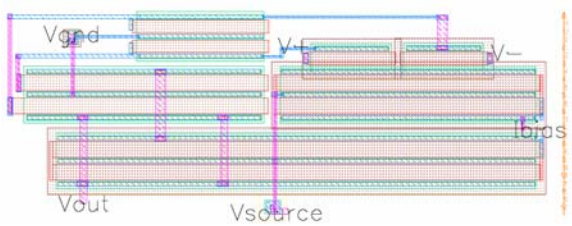


Figure 6 OTA Layout



Optimally sized PMOS transistors have been used as input devices by keeping large gate area to minimize the effect of flicker noise and at the same time not increasing input-referred noise indefinitely.

A complete schematic block diagram of the DVS simulation environment has been depicted in Fig. 7. The Driver circuit consisting of the power FET with large current-driving capacity has been shown. This driver circuit can be duplicated depending on the necessary driving current for the whole chip. Source follower structure has been used with a high pass-filter at the load of the driver which will provide more regulated output and removing higher order harmonic contents at the output.

The layout for the regulator was completed using the IBM 130nm design rules and has a area, excluding the PFET and capacitor, of 80um x 100um as seen in Fig. 8.

Figure 7 Linear Regulator Schematic

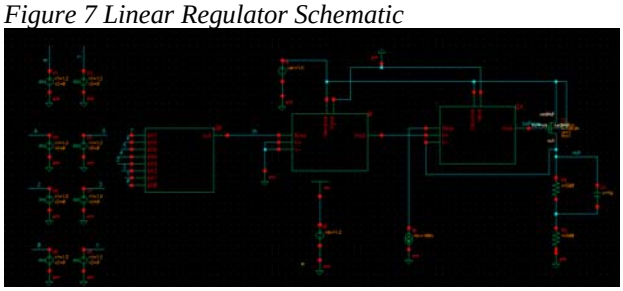
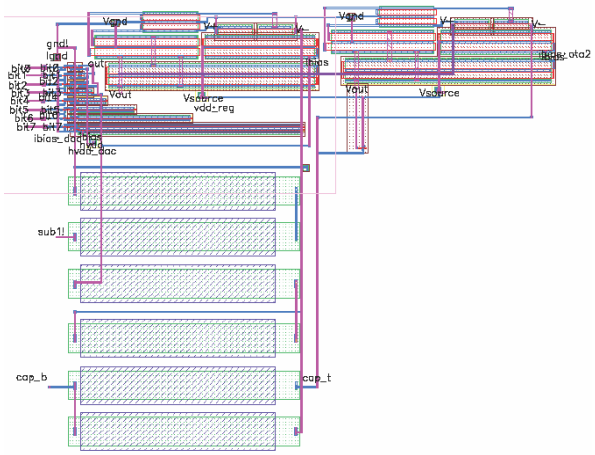


Figure 8 Linear Regulator Layout without power PFET and capacitor



V. Design for Test

The WIMS MCU itself has been tested and verified across two iterations of silicon. While the MCU does not have a scan-chain for debug, the data and address busses are available on several debug pins. Voltage and frequency signals relevant to this project can be brought out onto these pins. Additionally, three serial-to-parallel interfaces (SPI) will be available to communicate with the outside world.

For the digital-to-analog converter and digital filter, output will be taken out through an appropriately buffered analog pad. A bulk converter NMOS switch voltage can easily be tracked down to an off-chip test environment as it has large current carrying capacity. This will give a nice estimate of the frequency-to-voltage conversion mapping table. Corresponding frequencies will also be taken out and compared against digital input values.

VI. Results

The complete circuit has been designed and simulated in IBM 130 nm process and using Artisan standard cell library.

A. Dynamic Frequency Scaling

The counter based frequency divider has been simulated by setting the division number arbitrarily between 0 and 100.

The divider circuitry layout was placed and routed using Silicon Encounter as seen in Fig. 10. The divider consumes 24.7 µW with an area of 10794 μm^2. Clock buffer and driver power measurements are not included in this number.

Figure shows the operation of the frequency divided. When a byte is written into the counter register, the divider changes the clock frequency. In the transition from 0xOC and 0x07 there is a moment in which the clock pulses the minimal amount, which may be interpreted as a glitch in the hardware. Further control circuitry will be added to prevent this behavior from occurring.

Figure 9 Clock Divider Simulation Waveform

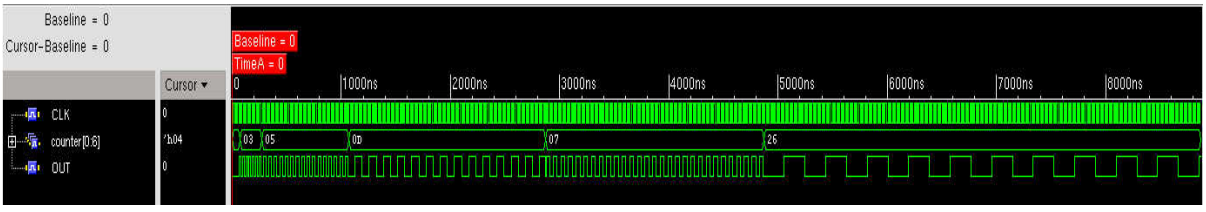
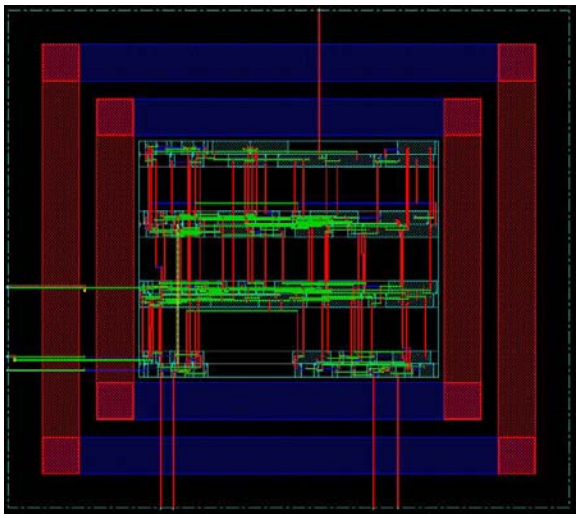


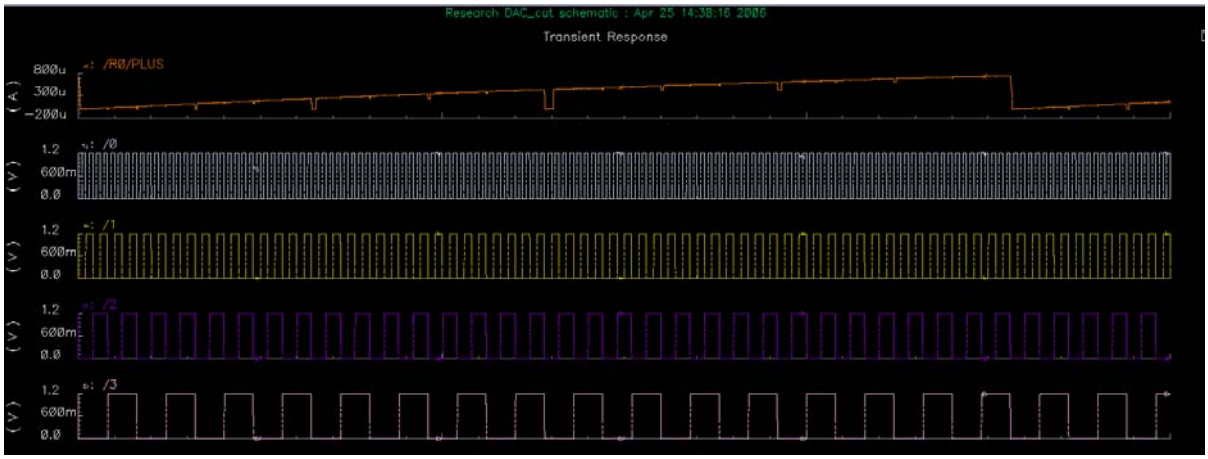
Figure 10 Clock Divider Layout



B. Dynamic Voltage Scaling

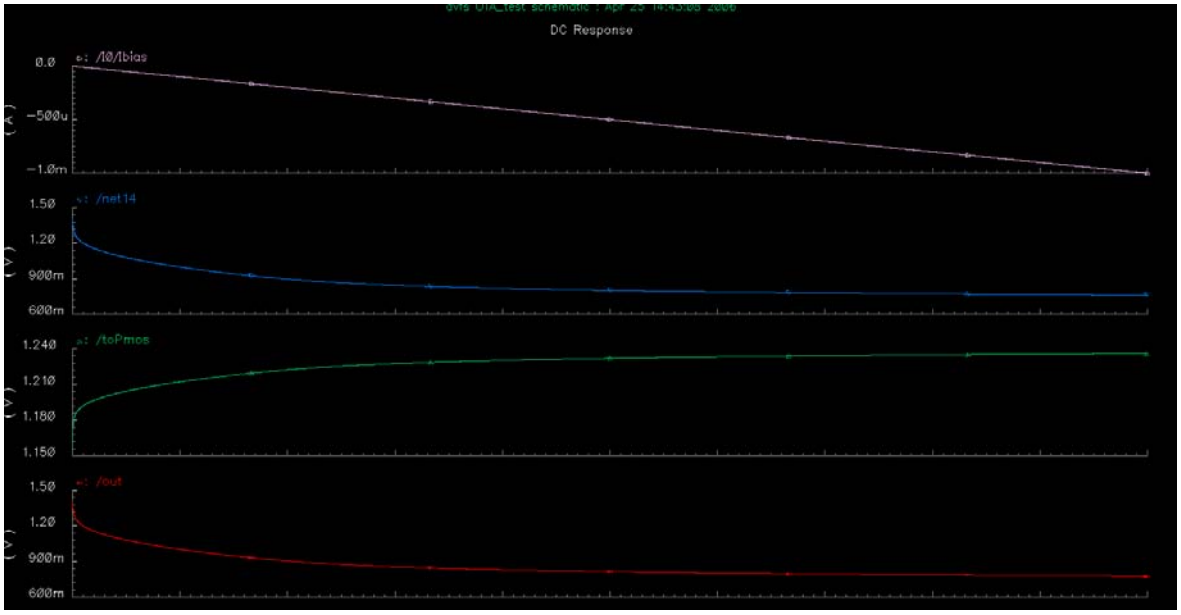
The 8-bit DAC circuit has been simulated using all possible bit patterns in the input. Correct output linearity is observed Fig. 11. Glitches observed in the waveform actually show correct functionality as designed and are a result of multiple input switching. Future work will focus on minimizing sharp transitions to maintain a consistent current output.

Figure 11 DAC Simulation



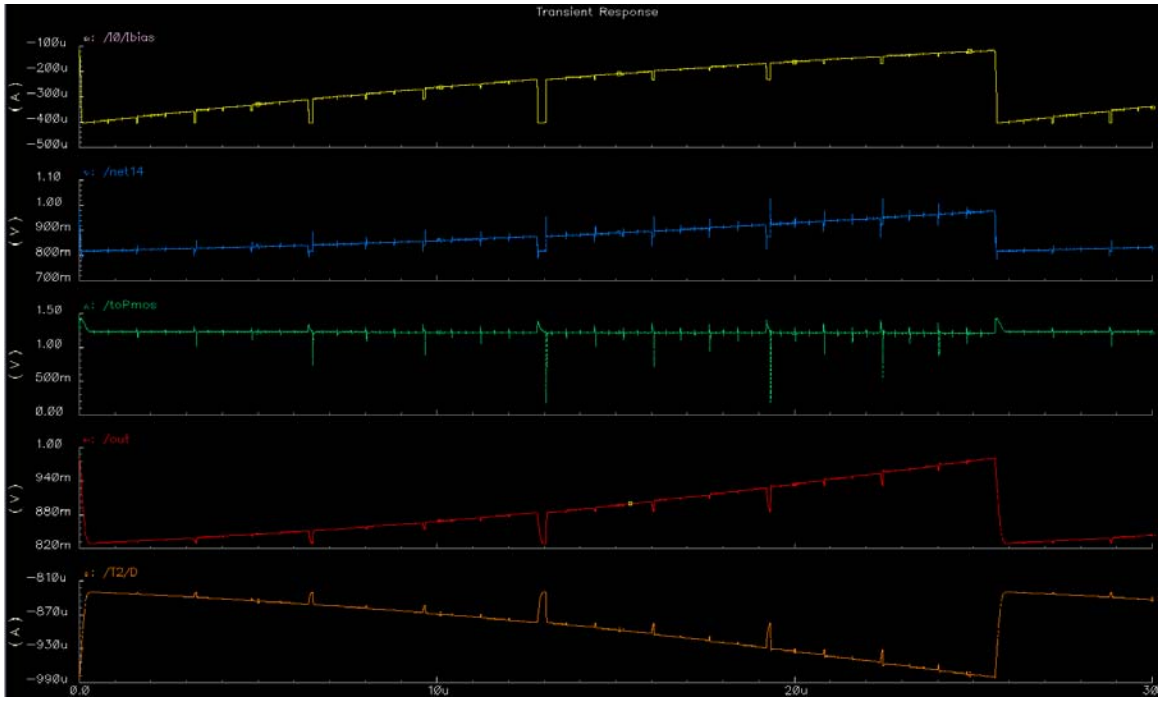
The linear voltage regulator has been simulated by changing the input current from 0 mA to 1mA linearly and output varies from 1.3 V to 0.8 V as shown in Fig 12. The output NMOS transistors enter are going to strong inversion region and makes the output saturated for more bias current. Hence our DAC output current will be scaled down using current mirror to match with the input current range of the linear voltage regulator. Fig. shows the voltage increase on the power PMOS, which changes current and the regulated voltage to the microcontroller.

Figure 12 Linear Regulator Spice Simulation



The complete DVS system simulation integrating all the blocks has been performed and output is seen to be linear during the range in which the system operates, Fig. 13. Currently enhancements are being made to increase the current range of the OTA and the linear regulator to allow more current to the core

Figure 13 Linear Regulator with DAC Spice Simulation



VII. Conclusions

The incorporation of dynamic voltage and frequency scaling into the WIMS microcontroller architecture will save significant power and energy. The circuits proposed in this project will allow for dynamically scaleable operation of a microcontroller. As the basic system has been designed over this course of this project, there remains as future work the actual integration of the DVFS system into the target microcontroller.

In addition to integration, several non-idealities currently exhibited by the DVFS system require further research. For example, the voltage does not scale as linearly as would be ideal, and the clock division circuitry passes some minimum-cycle "glitches" during transitions. Since much of our time and resources were spent on issues related to the IBM8RF technology, Artisan SAGE-X standard cell library, and the tools that can harness these resources, we left these issues as future work.

Beyond these improvements, future enhancements to the DVFS system include designing an optimal power PFET, decreasing power consumption across resistors used to stabilize the system, and optimizing the layout of the cells to minimize effects of process variation.

An important aspect of this project is the work involved in utilizing the Artisan library and corresponding tools for the IBM8RF technology. Manual layout was performed in the ibm-icfb environment following the 130nm design rules. Auto place-and-route layout was performed in SoC Encounter. Attached is a paper describing the steps to use these tools.

References

[1] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "A Dynamic Voltage Scaled Microprocessor System," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1571-1580.

[2] E. Marsman, *et al*, "A 16-bit Low-Power Microcontroller with Monolithic MEMS-LC Clocking," *International Symposium on Circuits and Systems*, May 2005.

[3] M. Weiser, B. Welch, M. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy," In *Proceedings of the First Symposium on Operating Systems Design and Implementation*, Monterey, CA, Nov. 1994, pp. 13-23.

[4] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor," In *Proceedings of the 7th Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.