

# 130nm Partially-Depleted SOI for Ultra Low Power Applications

*Abstract:* This paper focuses on the 130nm Silicon On Insulator (SOI) technology for Ultra Low Voltage and Power applications. A 16bit RISC microprocessor was considered for this purpose and the power consumption was evaluated using the 500mV, 130nm PD SOI technology. The power consumption was reduced using two techniques for power optimization - Clock Gating and operand isolation. Clock Gating involves disabling the clock when it is not required and Operand Isolation involves holding the input operands to a multiplexer or ALU stable when its output is not required. The power consumption of the 16 bit RISC microprocessor has been calculated for a simple program of Fibonacci series. Clock Gating on the register bank and isolation of operands involved in calculation of memory address for instruction fetch and jump cycles have been used to reduce the power consumption.

## 1. Introduction

Scaling down the device size leads to an increase of speed and also an increase of dynamic and static power consumption. There are a wide range of applications where the speed is not the main driver but Ultra Low Power (ULP) is a real issue (pacemakers, hearing devices, wearable devices, wrist watches etc). Achieving ULP and sufficient computing power is a conflicting requirement that can be trade-off by scaling down the power supply  $V_{dd}$  according to a given application. Reducing  $V_{dd}$  is very attractive because dynamic power is strongly reduced and given the fact that leakage currents are directly related to the electric field in the device, leakage power components are also drastically reduced. In this paper we deal with power supply of about 500mV targeting applications that need Ultra Low Power. The capability of SOI technology as a substitute to Bulk technology in applications at low voltage has been of interest in many applications [1].

Initially dedicated to military applications, SOI technologies have seen an increasing interest during the current decade thanks to a substantial progress in wafer fabrication. Many advantages are associated to SOI: good device

isolation, reduced parasitic capacitances, better dynamic device characteristics, good robustness against soft errors and finally better Ion/Ioff trade-off. Some drawbacks have been reported mainly for Partially-Depleted devices and are due to the floating body effect: kink effect, history effect, parasitic bipolar transistor and finally self heating. All these drawbacks are eliminated when running SOI circuits in ULV [2].

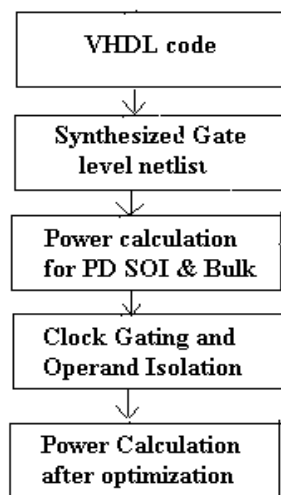
The aim of this paper is to analyze the performance of a ULV 16 bit RISC microprocessor and to use clock gating and operand isolation to minimize the power consumption for applications related to iris identification. Iris feature extraction using a wavelet packets algorithm has been presented earlier in [3]. Using the ULV RISC processor, an SoC for Iris identification is intended to be made operating with low power consumption. The algorithm for iris identification is given in [3, 4].

The architecture and the instruction set of the processor are presented along with the power consumption results of clock gating and operand isolation for both SOI and Bulk technologies as a comparison. The power has been estimated for both nominal voltage of 1.2V as well as for ULV of 500mV for both SOI and Bulk technologies.

## 2. Design Flow

The design flow involves the following steps as shown in Figure 1. First the code for the 16 bit RISC processor was written using VHDL with stress on coding style for low power as given in [5]. The architecture of the RISC processor is given later. Initially only a few instructions have been added in the code but the code will be extended for the application of Iris identification. The next step was to synthesize the code using Design Compiler. The synthesized gate level netlist was used to generate the activity file using a testbench. The testbench provides a clock and a reset signal.

The reset input is held active for the first 24ns and the clock cycle continues with period of 24ns. In case a sleep instruction is to be executed then a wake signal can be added in the testbench. The output of the ALU can be monitored at the *ALUOut* port. The annotated switching activity was used to estimate the power consumption using PrimePower. The power was estimated for nominal voltage of 1.2V and for ULV of 500mV. A reduction in power consumption was then attempted by introducing latch based clock gating and introducing combinational gates controlled by a signal to keep the operands at their previous values when they are not used for memory address calculation [6]. Power Compiler was used for the purpose of inserting clock gating and operand isolation.



**Figure 1: Design Flow**

The Clock Gating is used to deactivate the clock when it is not required by the register bank so that the power consumption is reduced. The Operand isolation prevents the inputs of the multiplexer from changing when its output is not required which again saves power. The operand isolation is performed on the step that calculates the target memory address to be written into the Program Counter.

The power optimization is achieved for the PD SOI ULV library. A significant saving in power is achieved over the nominal 1.2V library by using the ULV library with clock gating and operand isolation.

### 3. RISC architecture

The 16 bit RISC microprocessor that was coded in VHDL has the following specifications: -

1. 4 bit op-code for a maximum of 16 instructions
2. 16 bit register bank of 16 registers
3. 256\*16 bit memory space addressable using 8 bit address.
4. 16 bit ALU to perform the logical operations

The RISC microprocessor was made with the following instructions: -

1. Arithmetic & Logical – ADD, SUB, OR, AND, XOR
2. Memory related – Load from memory and Store to memory address
3. Immediate – Move immediate data
4. Conditional – Jump to memory address if condition is true
5. Unconditional Jump to a memory address to loop a program
6. Halt to stop the program
7. Sleep instruction

The RISC processor was tested using a program to generate the Fibonacci series. The sleep instruction is used to put the processor in a temporary state where neither the inputs or outputs of any of the registers or memory locations change. The clock cycle is prevented from reaching the processor by using a latch controlled by a signal which is generated when the instruction is executed. The processor comes out of the sleep state when an external wake signal is asserted.

The simulations were done for a total run-time of about 50 microseconds out of which the processor was idle after the execution of the program (to generate 10 Fibonacci series numbers) was completed at about 26 microseconds.

A clock cycle with time period of 24ns was used with a 50% duty cycle. The first state is the reset state when the external input reset goes high for the first 24ns. Then the next clock cycle is used to fetch the instruction from memory. The fetch is completed at the beginning of third clock cycle which is the op-code decode cycle. On the basis of the opcode appropriate control signals are issued in the

next clock cycle to begin the command execution.

The Logical commands take two more clock cycles to read the data, perform the calculation and then to store it in the register file.

However if the instruction is memory related then the first clock cycle after the opcode is decoded is used to calculate the memory address, the next to read the value to be loaded or stored & the final cycle to store in memory

or Register file. The conditional instructions check whether the contents of the specified register are zero and accordingly read the memory address to jump to. The bit-wise Instruction Set architecture and the processor architecture are shown on Figure 2 and Figure 3 [7, 8]. The difference between sleep and halt is that the processor can come out of the sleep state when an external wake signal is issued and continue its execution.

| S.No. | Instruction  | 15 | 14 | 13 | 12 | 11                            | 10 | 9 | 8 | 7                 | 6 | 5 | 4 | 3                      | 2 | 1 | 0 |
|-------|--------------|----|----|----|----|-------------------------------|----|---|---|-------------------|---|---|---|------------------------|---|---|---|
| 1     | ADD          | 0  | 0  | 0  | 0  | Source Reg 1                  |    |   |   | Source Reg 2      |   |   |   | Destination Reg        |   |   |   |
| 2     | SUB          | 0  | 0  | 0  | 1  | Source Reg 1                  |    |   |   | Source Reg 2      |   |   |   | Destination Reg        |   |   |   |
| 3     | AND          | 0  | 0  | 1  | 0  | Source Reg 1                  |    |   |   | Source Reg 2      |   |   |   | Destination Reg        |   |   |   |
| 4     | OR           | 0  | 0  | 1  | 1  | Source Reg 1                  |    |   |   | Source Reg 2      |   |   |   | Destination Reg        |   |   |   |
| 5     | XOR          | 0  | 1  | 0  | 0  | Source Reg 1                  |    |   |   | Source Reg 2      |   |   |   | Destination Reg        |   |   |   |
| 6     | Store Word   | 0  | 1  | 0  | 1  | Reg with Mem Address          |    |   |   | 4 bit Mem. Offset |   |   |   | Reg With Data to store |   |   |   |
| 7     | Load Word    | 0  | 1  | 1  | 0  | Reg with Mem Address          |    |   |   | 4 bit Mem. Offset |   |   |   | Reg to load data to.   |   |   |   |
| 8     | Jump address | 0  | 1  | 1  | 1  | Jump Address                  |    |   |   |                   |   |   |   | X                      | X | X | X |
| 9     | Move Imm.    | 1  | 0  | 0  | 0  | 8 bit data extended to 16 bit |    |   |   |                   |   |   |   | Destination Reg        |   |   |   |
| 10    | Jump on Zero | 1  | 0  | 0  | 1  | Jump Address                  |    |   |   |                   |   |   |   | Reg to check           |   |   |   |
| 11    | Sleep        | 1  | 0  | 1  | 0  | X                             | X  | X | X | X                 | X | X | X | X                      | X | X | X |
| 12    | Halt         | 1  | 1  | 1  | 1  | X                             | X  | X | X | X                 | X | X | X | X                      | X | X | X |

Figure 2: The Instruction Set architecture for the 16 bit RISC microprocessor

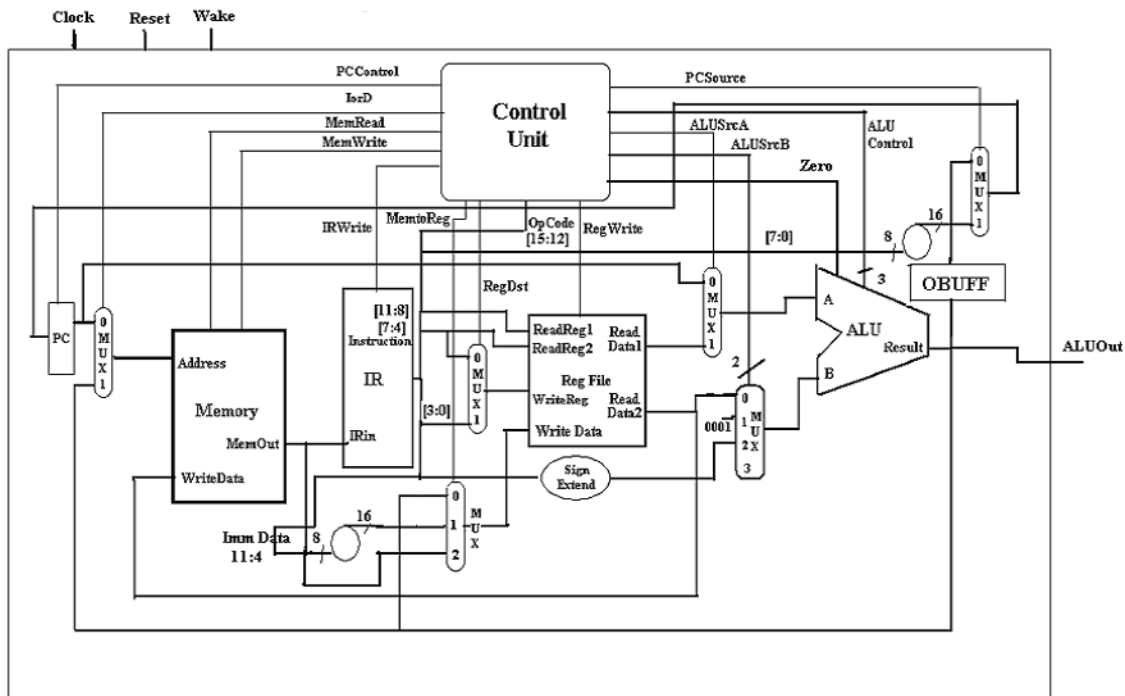
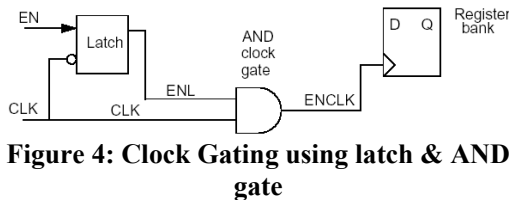


Figure 3: The 16 bit RISC processor architecture

#### 4. Power Estimation

The power consumed by the 16 bit RISC processor was evaluated using the 0.13 micron PD SOI library. A significant improvement was achieved with clock gating on the Register Bank, the output buffer register and the Instruction Register. Latch based gating was used along with AND gate and a control signal to enable the latch (Figure 4).



**Figure 4: Clock Gating using latch & AND gate**

The values of the power consumed are shown below for the various cases. A comparison between the Bulk & SOI technologies shows the advantage of SOI over the Bulk technology for ULV applications.

Table 1. shows the values for the ULV case when the 0.13 micron 500mV libraries were used. The values reported are without clock gating and operand isolation for both SOI & Bulk Si. Even though the power consumed by the Register File is more for the SOI technology than that consumed by the Bulk Si, the overall power consumption in the SOI 500mV case is much less as compared to the Bulk 500mV case. There is a power saving of about 16.25% with the SOI technology as compared with the Bulk technology.

Table 1. Power Consumption for Ultra Low Voltage SOI & Bulk Si based 16 bit RISC processor

| Technology used | Power consumption in Watt |           |           |           |  |           |
|-----------------|---------------------------|-----------|-----------|-----------|--|-----------|
|                 | Datapath + Memory         | Memory    | Reg File  | ALU       |  | Total     |
| 130nm 0.5V      |                           |           |           |           |  |           |
| SOI             | 5.307e-04                 | 4.702e-04 | 5.088e-05 | 3.025e-06 |  | 5.374e-04 |
| Bulk            | 6.337e-04                 | 5.777e-04 | 4.552e-05 | 3.721e-06 |  | 6.402e-04 |

Table 2. Power Consumption for SOI & Bulk Si based 16 bit RISC processor at nominal voltage

| Technology used | Power consumption in Watt |           |           |           |  |           |
|-----------------|---------------------------|-----------|-----------|-----------|--|-----------|
|                 | Datapath + Memory         | Memory    | Reg File  | ALU       |  | Total     |
| 130nm 1.2V      |                           |           |           |           |  |           |
| SOI             | 2.972e-03                 | 2.728e-03 | 2.021e-04 | 1.065e-05 |  | 3.007e-03 |
| Bulk            | 3.381e-03                 | 3.102e-03 | 2.314e-04 | 1.371e-05 |  | 3.418e-03 |

Table 2. reports the power consumption for the nominal voltage libraries of 1.2V. This helps in distinguishing the advantage of operating the transistors at ULV instead of at nominal voltage.

Thus it can be seen that by using the ULV SOI library we can achieve a power saving of about 82% over the nominal voltage of 1.2V. This significant saving in power offsets the decrease in performance as we scale down the voltage. It can be seen that most of the power is consumed by the Memory unit (more than 90% of the total power). Also it is seen that the power consumption increases in the Register file in the SOI technology as we scale down to

the ULV over the Bulk technology for the same case. Thus in order to save the power we have implemented clock gating over the register file thus disallowing the clock from reaching the registers when they are not being read. Also in order to save some power in the memory unit we have performed operand isolation over the operands that calculate the memory address thus holding the inputs of the multiplexer connected to the Program Counter stable when its output is not being used in the clock cycles where the contents of PC are not to be changed. The results of the power estimation are shown in Table 3. for the Ultra Low Voltage case.

Table 3. Power optimization for ULV SOI & Bulk Si based 16 bit RISC processor using clock gating and operand isolation

| Technology used                                    | Power consumption in Watt |           |           |           |  |           |
|--|---------------------------|-----------|-----------|-----------|--|-----------|
|  | Datapath + Memory         | Memory    | Reg File  | ALU       |  | Total     |
| 130nm 0.5V with clock gating and operand isolation |                           |           |           |           |  |           |
| SOI  | 4.919e-04                 | 4.571e-04 | 3.082e-05 | 7.564e-07 |  | 4.957e-04 |
| Bulk   | 6.130e-04                 | 5.715e-04 | 3.695e-05 | 6.830e-07 |  | 6.164e-04 |

By using the clock gating and operand isolation the power saving in PD ULV SOI over Bulk Si for  $V_{dd} = 500\text{mV}$  has increased to almost 20% and over the nominal voltage it has increased to about 84%. Thus it can be seen that ULV SOI is a much favourable candidate for low voltage low power applications.

The Figure 5. shows the improvement in power saving when we use the nominal voltage for the processor & when we use the Ultra low voltage SOI technology with clock gating and operand isolation

A comparison between SOI and Bulk technologies can also be made from the graph.

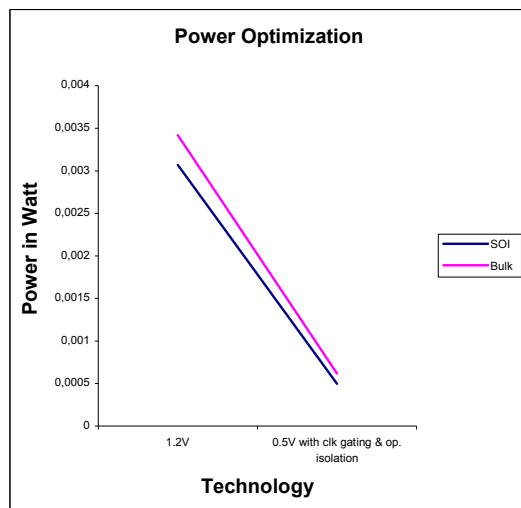


Figure 5: SOI vs. Bulk for ULV applications

## 5. Iris Identification Application

### 5.1. Overview

In applications where power consumption is more an issue than speed the SOI 500mV technology can be useful. The techniques of clock gating and operand isolation also help in optimizing the power.

One such application that involves the use of such a low power microprocessor is the SoC for Iris identification as mentioned in [4]. This paper presents a biometric technology for identifying people based on iris. An algorithm using wavelet packet analysis is presented. The complete implementation overview is presented on Figure 6 for such applications.

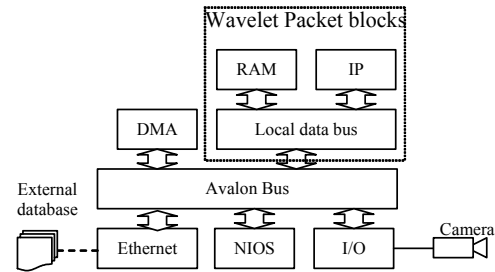


Figure 6: implementation overview

At this time, the processor, called NIOS from Altera Inc., is used to read the image acquired by the camera, to extract and to unwrap the iris and to send it to a wavelet packets IP. After applying the wavelet packets transformation, interesting packets are exploited by the NIOS to generate a signature. Thus, the two main components are the processor and the wavelet packets IP.

### 5.2. Wavelet packets architecture

Figure 7 describes the wavelet packets architecture.

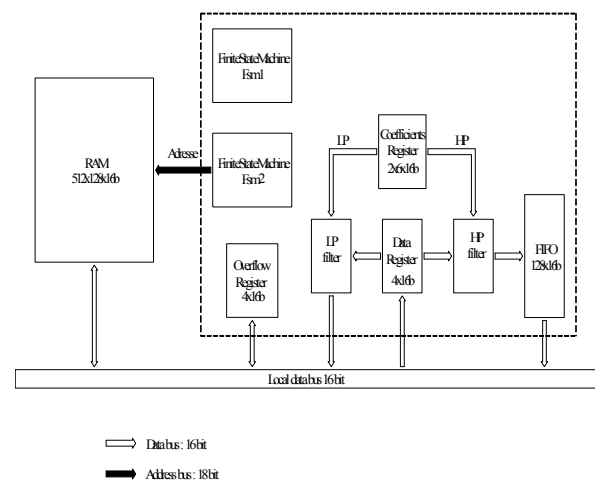


Figure 7: Wavelet packets architecture

The original image is located into an external RAM, and we have chosen an algorithm that works only with one memory bus. Thus, we organize the processing to get one RAM access per clock cycle.

The first level of decomposition (applied on the complete 2D image) requires one full read and one full write for lines processing, and one full read and one full write for columns processing. For a 512 by 128 pixels image, it

requires approximately 256 kclock cycles. The size of unwrap iris image is a parameter and could be change for any unwrap image size (power of 2). We set by default the maximum size to 512x128 pixels. Each pixel is extended to 16-bit length because 16-bit operation is enough to give a good accuracy for wavelet decomposition coefficients.

The filters contain 6 parallel MACs (Multiplier-Accumulators). The products of multiplication are accumulated and the final result is truncated to 16 bits. The LP filter output is connected to local data bus because the result of this filtering is stored directly into RAM, at the already read pixel location. The HP filter output could not be stored into the RAM because it will erase a non-read pixel. The result of this HP filtering is stored into a FIFO.

To down-sample the input signal, filters are shifted by two pixels at each new processing step. Thus, four pixels are common to two following processing step: instead to read again four pixels from RAM, these four pixels are stored into the data register.

The coefficients register stores LP and HP filters coefficients. The FIFO memory stores data coming from HP filter processing till the location where they should write into RAM is available. For an image size of 512x128 pixels, its size is set to 128 words because writing into RAM memory could start when filters reach half image size. The overflow register is used to manage boundary effects. The first 4 pixels are stored inside this register and are used at the end of the processing. Two state machines manage the FWT processing.

### 5.3. Synthesis results

The wavelet packets IP is synthesized on Bulk and SOI technologies at  $V_{dd} = 500\text{mV}$  and the results presented on Figure 8 show that SOI technology dissipates 16% less dynamic power than bulk technology for the wavelet packet IP.

|      | P <sub>dyn</sub> ( $\mu\text{W}$ ) | P <sub>leak</sub> ( $\mu\text{W}$ ) |
|------|------------------------------------|-------------------------------------|
| Bulk | 11.39                              | 59.2                                |
| SOI  | 9.53                               | 22                                  |

**Figure 8: Dynamic & Leakage Power Dissipation**

## 6. Conclusion and future work

The RISC microprocessor that has been discussed here for shows nearly 20% savings in dynamic power over the Bulk Si technology with power optimization. In portable applications where the life of the battery determines the feasibility of the application, it becomes imperative to try and optimize power as much as possible even if it is at a slight disadvantage to the performance in terms of speed. The speed is not a critical issue in the processor developed & hence we have used a clock period of 24ns.

Future work involves using such Ultra Low Power processors as presented here and combining it with the wavelet packets IP to realize our iris identification application on a SoC. The wavelet IP currently simulated with the NIOS processor IP shows a 16% saving while the power optimization applied on SOI 500mV library with the RISC processor shows a 20% saving. This processor is intended to be used in this application with the wavelet packet IP further.

## 6. References

- [1] Akihiro Ebina, "Ultra-low power CMOS IC using partially depleted SOI technologies", EPSON, SOI workshop Paris France.
- [2] Olivier T., "SOI Partially-Depleted Ultra Low Voltage Memory and Digital Circuit Design", ICIDT 2005.
- [3] Rossant F. et al, "IRIS identification and robustness evaluation of a wavelet packets based algorithm", ICIP 2005, September 2005, Belgium.
- [4] Ea T. et al, "Implementation on SoPC of algorithms dedicated to Iris identification" DCIS 2005, November 2005, Portugal
- [5] "Low Power Electronics Design", Christian Piguet, CRC Press.
- [6] Synopsys Power Compiler & Prime Power User guide.
- [7] "Computer Organization and design" Patterson D., Hennessey J., 2<sup>nd</sup> Ed. M.Kaufmann
- [8] "Kraken" a 16bit RISC processor developed by Kim D., Lin A., Stanford University, 2000.
- [9] Rydgren E. et al, "Iris Features Extraction Using Wavelet Packets", ICIP 2004, October 2004, Singapore.