

## Low power implementation of datapath using regularity\*

LAI Li-ya (赖莉雅)<sup>†</sup>, LIU Peng (刘鹏)

(Department of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310027, China)

<sup>†</sup>E-mail: lailiya@etang.com

Received Apr. 12, 2004; revision accepted Aug. 18, 2004

**Abstract:** Datapath accounts for a considerable part of power consumption in VLSI circuit design. This paper presents a method for physical implementation of datapath to achieve low power consumption. Regularity is a characteristic of datapath and the key of the proposed method, where synthesis is tightly combined with placement to make full use of regularity, so that low power consumption is achieved. In This paper, a new concept of Synthesis In Relative Placement (SIRP) is given to deal with the semi-regularity in some datapath. Experimental results of a sample circuit validated the proposed method.

**Key words:** Datapath, Low power, Regularity, Relative placement

**doi:**10.1631/jzus.2005.A0596

**Document code:** A

**CLC number:** TN391.72

### INTRODUCTION

Power consumption has become an ever increasing crucial issue in VLSI circuits design due to the wide use of portable and high-density micro-electronic devices. Battery life, packing, cooling and reliability are all issues that relate to power consumption. Datapath consists of functional units, memory or storage elements, and interconnection units implementing data transfer among functional units (Wayne, 2002). Well designed datapath can reduce power consumption considerably and greatly influence circuit performance.

Many low power design methods and technologies such as gating techniques (Chaeryung *et al.*, 1999) and several algorithms which perform scheduling, allocation and hardware partitioning for synthesis (Chiou *et al.*, 2001; Raghunathan and Jha, 1994) have been developed over the past years. However, they were mostly focused on the behavioral level or synthesis level. In this work, both synthesis and placement were taken into account during the designing of datapath. An approach that takes advantage of

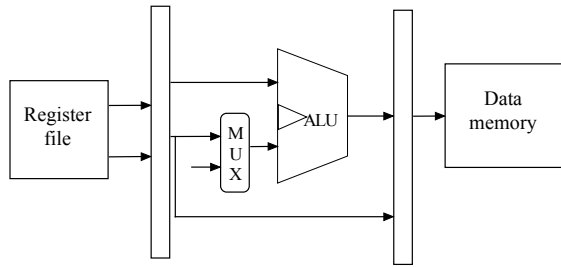
datapath's regularity to achieve low power consumption and other improvements in performance is presented.

### REGULARITY OF DATAPATH

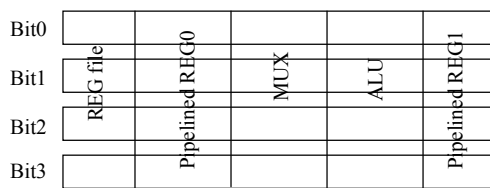
Fig.1 shows a typical datapath in a pipelined circuit. The data read from register file is operated in ALU, which may include adder, multiplier and shifter. The results from ALU are stored into memory. There are two banks of pipeline registers that divide the circuit into three stage pipelines as well as several multiplexer (MUX) for data selections.

Regularity of datapath means every bit in the same datapath has the same architecture; most significant bit (MSB) and least significant bit (LSB) in a datapath may have the same operations and are selected by the same MUX. One single bit called bit-slice can be built and duplicated several times to construct a completed datapath. Via this method, the datapath designer designing a 32-bit processor can concentrate on designing the bit-slice. Fig.2 presents four bit-slices constructing the datapath shown in Fig.1; in this case, each bit is arranged horizontally, while functional units are placed vertically.

\*Project (No. 2002 AA1Z1140) supported by the Hi-Tech Research and Development Program (863) of China and the Fork Ying Tong Education Foundation (No. 94031), China



**Fig.1 Datapath in a three-stage pipelined circuit. ALU operates on the data from the register file, and then writes the results into data memory**



**Fig.2 Four bit-slices of the datapath shown in Fig.1. Each bit-slice is placed horizontally, and the four bit-slices form a data flow from left to right. The functional units have vertical direction**

## LOW POWER ANALYSIS

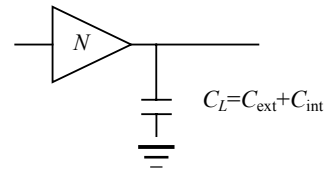
Power dissipation in CMOS circuits are caused by switching currents, short-circuit currents and leakage currents (Rabaey, 1996), of which leakage currents comprise the main consumer of static power; while the other two must be considered in the case of dynamic power consumption analysis. Although static energy consumption will get more and more attention when feature size decreases to below 0.1 micro (Kim *et al.*, 2003), dynamic power consumption is still an important issue in current chip design. Here is the expression for the dynamic power consumption:

$$P_{\text{dyn}} = C_L V_{DD}^2 f \quad (1)$$

where  $C_L$  is the load capacitance;  $V_{DD}$  is the supply voltage; and  $f$  is the frequency of switching activity. From Eq.(1), it is easy to draw the conclusion that decreasing capacitance, supply voltage and switching activity can reduce the power dissipation of the circuit. The  $V_{DD}$  factor is obviously the most influential because of the quadratic dependence. Many articles

have discussed switching activity reduction methods such as functional selection and clock gating. The proposed approach of reducing power consumption targets capacitance factors.

Fig.3 shows the load capacitance of an inverter whose size is  $N$  times that of minimum size one. Load capacitance  $C_L$  consists two elements: intrinsic capacitance ( $C_{\text{int}}$ ) and extrinsic one ( $C_{\text{ext}}$ ).  $C_{\text{int}}$  represents the intrinsic capacitance such as diffusion capacitance;  $C_{\text{ext}}$  stands for wiring capacitance and fan-out capacitance. Intrinsic capacitance of one gate is proportional to its area size. If one inverter occupies  $N$  times area of a minimum size inverter, its intrinsic capacitance will be correspondingly equal to  $N$  times that of the minimum size inverter.



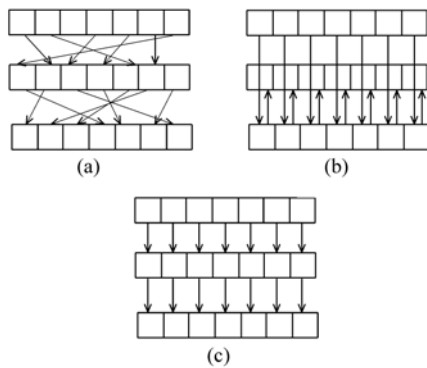
**Fig.3 Load capacitance of an inverter whose size is  $N$  times that of the minimum size inverter. Total load capacitance is the sum of intrinsic capacitance and extrinsic capacitance**

The above analysis points to several avenues for reducing load capacitance, namely, reducing the diffusion capacitance, devices size, wire length and fan-out capacitance.

## USING REGULARITY TO REDUCING CAPACITANCE

To make the best of the datapath regularity, it is very necessary to fully explore the usage of bit-slice. Fig.4a presents a datapath without properly arranged bit-slices. Though functional units are placed horizontally, every cell in the units has random location. Each bit shown in Fig.4b is arranged vertically, but the placement does not follow the correct data flow. Fig.4c shows a typical placement of datapath with bits aligned vertically and function units aligned in the horizontal direction. Comparison of Figs.4a, 4b and 4c shows that total wire length in Fig.4c is shorter than that in Fig.4a or Fig.4b. More conclusions can be drawn as follows after analysis:

1. Tiling bit-slices can substantially shorten wire



**Fig.4 Three different methods for the placement of datapath. (a) Placement without bit-slices; (b) Placement without correct data flow; (c) Placement with bit-slices and correct data flow**

length, especially for large width datapath. The wire length from most significant bit to least significant bit can reach 322  $\mu\text{m}$  in the worst condition for a 64-bit datapath under 0.18  $\mu\text{m}$  technology, so tiled datapath can lead to shorter wire length, and thus reduces the extrinsic capacitance part of load capacitance.

2. In Deep Submicro (DSM) VLSI designs, interconnection delay has significant effect on the overall performance. In 0.18  $\mu\text{m}$  technology, the undue delay caused by wire length is more than 50 percent of all path delays, more than 70 percent in the case of 0.13  $\mu\text{m}$  technology (Aart, 2002). By employing the bit-slice method, the reduction in wire length not only reduces the load capacitance, but also leads to better performance at higher frequency. As a result, smaller size cells can be used to interconnect these wires while maintaining the same wire delay. In some circumstances, some cells such as buffers can even be cut off without sacrificing the performance of the whole path. Obviously, few cells in the design result in less power consumption.

3. Because datapath's structure is regular, compact placement can be realized when bit-slices are properly tiled. However, existing CAD tools are poor at identifying and extracting the bit-slice structure of datapath, so automated placement usually results in unsatisfactory solutions with large area and low area utility. When tiling bit-slices, designers can make full use of space during placement with the information of regularity, which will lead to smaller area, and thus reduce diffusion capacitance—another part of the load capacitance.

When designing datapath, regularity can be used during synthesis and placement to decrease load capacitance in several ways, including shortening wire length, reducing the cell size or cell number, placing cells closely, and so on.

## SYNTHESIS IN RELATIVE PLACEMENT

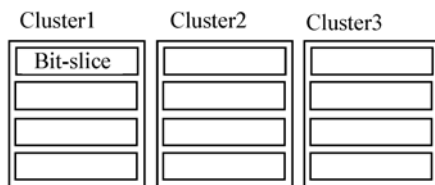
The above analysis shows the importance of regularity to low power implementation of datapath. How to use regularity is a problem. Many articles have discussed ways to choose the proper design style: custom design or application specific integrated circuit (ASIC) flow design using standard cells (Reinhardt, 2003; Chinnery and Keutzer, 2000; Daily and Chang, 2000). Due to the lack of experienced custom designer and some other reasons, ASIC flow design was chosen for the proposed datapath design. Though several papers pointed out how to use regularity in datapath design, they are concerned with synthesis or placement individually (Chowdhary and Gupta, 2002; Tao and De, 2000). However, if datapath with regular construction is synthesized while its location is still unknown, satisfactory result cannot be obtained. Synthesis should be tightly combined with placement in order to obtain better performance.

Ye *et al.* (2002) pointed out that many datapath circuits have the feature of semi-regularity meaning that the bits in a datapath circuit are not always identical; they may have differences in some functional units. This paper proposes a new method called Synthesis In Relative Placement (SIRP). This method first defines the relative locations of cells instead of fixed locations; then it performs logic synthesis while taking the information of relative placement obtained from the first step into account. As a result, shorter wire length, minimum sized cells and smaller area can be obtained to reduce power consumption compared with the cases without employing this method.

## Regularity extraction

The first step in using regularity of datapath is extracting regularity out of datapath, which is complicated, because of the existence of some differences between bits. It is easier to obtain the pattern of regularity at higher level rather than from synthesized

netlist. When defining the architecture of a datapath, the designer should bear in mind which aspects of regularity can be used during placement. When moving to the behavioral description (VHDL or Verilog) of the datapath, the designer must maintain such regularity. In a hierarchy design, not only must every bit in a module have similar structure, each module in the design must also have regular architecture. These modules are called clusters in the physical hierarchy, as shown in Fig.5. The main work of regularity extraction in a hierarchy design is the correct partitioning of the design into modules; as a result, bit-slices in each of the modules can be built and maintained in a similar structure.



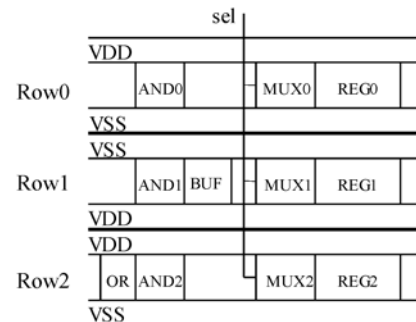
**Fig.5 Three clusters in a physical hierarchy design. Each cluster has four bit-slices and presents similar structure**

### Relative placement

In ASIC flow design, standard cells are placed restrictedly in rows which have the same orientation as that of the power supply. To take such limitations into account, bit-slices are placed horizontally so as to achieve shorter wire length and higher area utility. Instead of fixing each location of cells in a bit-slice, only relative locations are defined because of the datapath semi-regularity. As shown in Fig.6, three bit-slices are placed horizontally in three rows, each has the same function units such as AND, MUX and REG. The relative locations of cells in these same function units are defined. For example, AND0 has a location above AND1 and on the left of MUX0. Due to the datapath semi-regularity, not all bit-slices are identical. For example, as shown in Fig.6, slice in row2 has OR function unit which is different from that in other slices, and its location is not defined. Moreover, there exist control wires and buffers to interconnect wires, so some area is left for routing wire and placing buffer.

Not only must individual cells be defined with relative locations, but also in hierarchy design, modules called clusters in physical layout must have relative placements. As discussed above, the design is

partitioned into modules and regularity is extracted from these modules. So the designer can take each module as a cluster and define its relative location when placing design circuits to achieve better performance.



**Fig.6 Layout of three bit-slices. VDD, VSS stand for power supply in each channel. The relative location of functional units of AND, MUX and REG are defined, but OR and BUF are left alone. Some area is left for control net (sel) routing and buffer placing**

### Incremental synthesis

After the information on relative locations of cells or clusters is obtained, the designer can optimize synthesis in relative placement. Traditional synthesis without placement information uses wire load model to estimate delays, which may lead to pessimistic estimations of performance, and to use many buffers and larger size cells that will result in higher power consumption. With the proposed SIRP method, more accurate delay information can be obtained after relative placement. As a result, more suitable cells will be used during synthesis. For instance, the buffer in Fig.6 may be no longer needed and the size of MUX can be smaller because of relative placement. All of these will benefit low power circuit design.

### EXPERIMENTAL DETAILS

This paper focuses on synthesis and placement of datapath during ASIC design flow. The datapath modules are described as Register Transfer Level (RTL) in Verilog language, and synthesized into netlist using Design Compiler. After that the synthesized netlist is sent to Astro for floorplan. Then both the floorplan information and the definition of cells' relative location are used in Physical Compiler for placement. At last some adjustments for clusters and incremental synthesis will be performed for getting

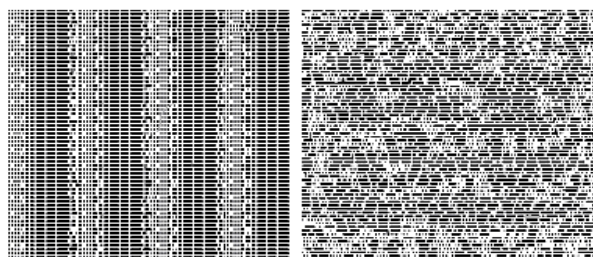
better results.

Experiments were performed on MDS register file using SMIC 0.18  $\mu\text{m}$  technology libraries. MDS register file is an 8-word/64-bit register file used in multimedia data stream processing. The results were compared with those of a traditional ASIC flow design. The comparison included number of nets, total net length, average net length, special nets—clock and reset nets, number of cells, cell area, module delay and dynamic power consumption.

Table 1 compares the results of SIRP and that of traditional ASIC flow. SIRP reduces the number of nets by 10.78%, total net length by 13.30%, and average net length by 2.80%. The most significant nets were the net of clock and reset, both of which were reduced by more than 58% due to regular placement. SIRP also reduced the number of cells by 12.80% and cell area by 14.29%, which means less buffers needed to interconnect wires and more compact placement. The reduced wire length and cell area also reduced dynamic power consumption by 27.52% even in the condition of improvement of timing. The layouts of MDS register file with and without SIRP are shown in Fig.7.

**Table 1 The comparison between the results of SIRP and traditional ASIC flow**

	Traditional flow	SIRP	Improvement
Net number	3972	3544	10.78%
Total net length ( $\mu\text{m}$ )	612720.75	531351.92	13.30%
Ave net length ( $\mu\text{m}$ )	154.26	149.93	2.80%
Net clock length ( $\mu\text{m}$ )	6842.54	2842.85	58.83%
Net reset length ( $\mu\text{m}$ )	6866.7	2816.84	58.98%
Cell number	3390	2956	12.80%
Cell area ( $\mu\text{m}^2$ )	107326.30	91984.94	14.29%
Delay read/write (ns)	1.4975/ 0.9932	1.3040/ 0.9859	12.92%/ 0.73%
Power (mW)	77.8764	56.4424	27.52%



**Fig.7 Left is the layout picture of MDS register file with SIRP, and right is result of traditional ASIC flow design**

## CONCLUSION

This paper describes a new method for low power physical implementation of datapath. The key of this method is the feature of datapath—regularity. A new concept of Synthesis In Relative Placement (SIRP) is given to deal with the semi-regularity in some datapath. At last, an experiment in a concrete chip design proved that such method can significantly reduce power consumption and has many other benefits such as reduced area, improved timing for design performance, etc.

## References

- Aart, D.G., 2002. Trust, Technology, & Momentum. [http://www.synopsys.com/corporate/exec\\_presentation/2002/IEF\\_arrt.pdf](http://www.synopsys.com/corporate/exec_presentation/2002/IEF_arrt.pdf)
- Chaeryung, P., Taewhan, K., Liu, C.L., 1999. An Integrated Approach to Data Path Synthesis for Power Optimization. The 12th Annual IEEE International Conference on ASIC/SOC, Washington, DC, USA, p.125-129.
- Chinnery, D.G., Keutzer, K., 2000. Closing the Gap Between ASIC and Custom: An ASIC Perspective. DAC'00, Los Angeles, CA, p.637-642.
- Chiou, L.Y., Muhammand, K., Roy, K., 2001. DSP Data Path Synthesis for Low-power Applications. ICASSP'01, Salt Lake City, UT, USA, p.1165-1168.
- Chowdhary, A., Gupta, R.K., 2002. A methodology for synthesis of data path circuits. *Design & Test of Computers*, **19**(6):90-100.
- Daily, W.J., Chang, A., 2000. The Role of Custom Design in ASIC Chips. DAC'00, Los Angeles, CA, p.643-672.
- Kim, N.S., Austin, T., Baauw, D., Mudge, T., Flautner, K., Hu, J.S., Irwin, M.J., Kandemir, M., Narayanan, V., 2003. Leakage current: Moore's Law meets static power. *Computer*, **36**(12):68-75.
- Rabaey, J.M., 1996. Digital Integrated Circuits: A Design Perspective. Prentice-Hall, New Jersey.
- Raghunathan, A., Jha, N.K., 1994. Behavioral Synthesis for Low Power Computer Design. ICCD'94, Cambridge, MA, USA, p.318-322.
- Reinhardt, M., 2003. Closing the Gap Between ASIC and Full Custom: A Path to Quality Design. The 4th International Symposium on Quality Electronic Design, p.255.
- Tao, Y.T., De, M.G., 2000. Data Path Placement with Regularity. ICCAD'00, San Jose, CA, USA, p.264-270.
- Wayne, W., 2002. Modern VLSI Design: System-on-Chip Design, 3rd. Prentice-Hall, New Jersey.
- Ye, T.T., Chaudhuri, S., Huang, F., Savoj, H., De, M.G., 2002. Physical Synthesis for ASIC Datapath Circuits. ISCAS 2002, Phoenix, USA, p.365-368.