

# Power Estimation in Digital CMOS VLSI Chips

Duong Tran Kyung Ki Kim Yong-Bin Kim, *Member, IEEE*

**Abstract**—Power estimation method of digital CMOS VLSI chips is proposed in this paper. The chip is divided into five sections, which consists of logic circuit, on chip memory, interconnection, clock distribution, and off chip driving (I/O), and the power consumption of each part is estimated separately. In addition, an efficient method for estimating gate-count from Verilog register transfer level (RTL) descriptions is developed, and implemented into an estimation tool written in C language. The proposed estimation method contributes to investigate power distribution among different parts of a digital CMOS VLSI chip and to provide guidelines for early design stage.

**Index Terms**—power estimation, logic circuit, on chip memory, interconnection, clock distribution, I/O, Verilog

## I. INTRODUCTION

REDUCING power consumption in CMOS VLSI chips becomes increasingly important due to increase in circuit speed and density. Therefore, it is essential to obtain overall idea of controlling parameters that contributes to dominant power consumption in different sections of the chip in early design phase. Power estimation method is analyzed for various components of digital CMOS VLSI chips such as, logic circuit, memory, interconnection, clock distribution, and off chip driving (I/O). The gate-count estimation from Verilog RTL descriptions is also developed. Estimation methods are implemented into a tool using C language, which takes a Verilog description file as an input and outputs the number of logic gates required to implement the design. The Verilog design descriptions from [10], [11], and [14] are employed for testing and the gate count estimated tool provides a proximate number of logic gates needed to implement those Verilog design descriptions.

## II. MEMORY ON CHIP

The Fig. 1 shows a basic structure of a SRAM. The power consumption of on chip memory can be represented as the sum of power consumption of each part.

### A. Decoders

The X-Decoder consists of decoding part and the driving part. The decoding part can be regarded as a random logic block and the power consumption is highly dependent on the type of logic blocks used to implement the decoder, which can be described by activity factor. Activity factor is 0.25 for a static CMOS design, and 0.5 for

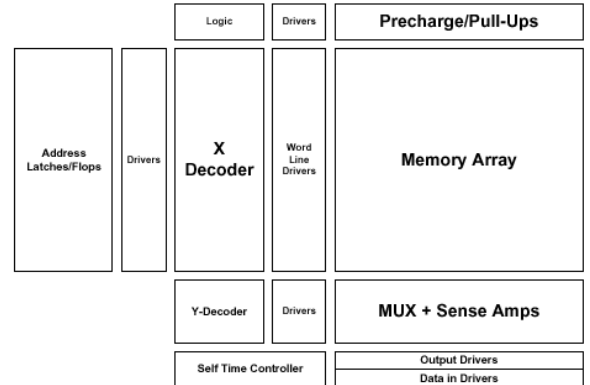


Fig. 1. Sram Basic Structure

a precharge/discharge circuit. The driving part contains drivers that drive outputs of the logic circuit block. Thus, the power of X-Decoder is the sum of the power of the input Latches/Flops, Logic part, and output drivers:

$$P_{x-decoder} = P_{input-latch} + P_{logic} + P_{drivers} \quad (1)$$

The power consumption in Y-Decoder is similar to that of X-Decoder, however, few things has to be taken into account. The Y-Decoder contains the high order bits of the memory address, and it is likely that this decoder will drive the same address during the time with sequential memory address input, unless the whole address is changed. If it is not clocked, the decoder changes its address state non-sequentially and for the rest of the time, the activity factor can be assumed to be zero.

### B. Bit Lines and Bit Lines Precharge Devices

Every time there is either a READ or WRITE cycle in the array, bit lines are partially discharged. The difference between READ and WRITE is the voltage to which the bit lines are discharged. The READ or WRITE recovery devices have to restore the bit lines to their standby voltage, which can be lower than  $V_{dd}$ . Due to the fact that in some SRAMs several columns are connected to one sense amplifier (sense amp) through a mux, the bits behave differently during READ and WRITE. During READ those columns that are connected through the mux to the sense amp are discharged to a different voltage than those that are blocked by the mux. If the muxes are X to ONE type, then the number of columns that are connected to the sense amps is:

$$Y = \frac{\text{Number of Bits}}{X} \quad (2)$$

Thus, the average bit line recovery current for columns connected to sense amps is:

$$I_{avg1} = P_{read} \left[ Y (V_{init-read} - V_{f-read-1}) C_{bf} \right] \quad (3)$$

Department of Electrical and Computer Engineering  
Boston, MA, 02115, USA  
Phone: (617) 373-7780, email: dtran@ece.neu.edu

Department of Electrical and Computer Engineering  
Boston, MA, 02115, USA  
Phone: (617) 373-7780, email: kkim@ece.neu.edu

Department of Electrical and Computer Engineering  
Boston, MA, 02115, USA  
Phone: (617) 373-2919, email: ybk@ece.neu.edu

Where,  $P_{read}$  is probability for a read cycle.  $V_{f-read-1}$  is the lowest voltage the bits that are connected to the sense amps can reach.  $V_{init-read}$  is the voltage on the bit line BEFORE start of READ.  $C_b$  is capacitance of ONE bit line. The number of columns that are not connected to the sense amp during READ cycle is: Number-of-Bits -  $Y$ . Thus, the average bit line recovery current for columns not connected to sense amps is:

$$I_{avg2} = P_{read} \left[ (N_b - Y)(V_{init-read} - V_{f-read-2})C_b f \right] \quad (4)$$

Where,  $N_b$  is number of bits in the array.  $V_{f-read-2}$  is the lowest voltage the bits that are NOT connected to the sense amps can reach. Using the same rational the WRITE cycle currents can be calculated where the final voltages on the bit lines are different, the probability to have a WRITE cycle and the number of bits written are accounted for.

The bit line precharge devices that are toggled every clock cycle have activity factor  $A = 1$ . If there are 2 precharge devices and one equalizer per bit and  $\bar{b}it$ , then the precharge devices average current required to drive their gate is:

$$I_{pre-devices} = \left[ (C_{equ} + 2C_{pre})N_{bit} + C_{line} \right] fV_{dd} \quad (5)$$

Where,  $C_{pre}$  is gate capacitance of ONE precharge device,  $C_{equ}$  is gate capacitance of the equalizer device, and  $C_{line}$  is total line capacitance routed between the precharge devices.

### C. Static Pull-Ups

Static pull-ups, if exist, are normally on during READ and off during WRITE. If static pull ups are toggled by a driver, the power can be calculated as for the gates of the bit line precharge devices.

### D. Sense Amps and Output Drivers

There are many types of sense amps. Some are pure dynamic and do not consume static power, while others contain current sources and consume static current.

$$I_{sense} = I_{static} * N + C_{sense} * N * 0.5 * f * V_{dd} \quad (6)$$

Where,  $I_{static}$  is static current consumed by the sense amps,  $N$  is number of sense amps, and  $C_{sense}$  is capacitance of the sense. The average current associated with the sense amp static output drivers is:

$$I_{drivers} = 0.25 * N * C_{drivers} fV_{dd} \quad (7)$$

Where,  $N$  is number of drivers and  $C_{driver}$  is average load on the output of the drivers.

### E. Memory Array

During READ cycle the static pull-ups (if present) supply DC current that has to be simulated. The average READ current in the array is:

$$I_{avg-mem-array-read} = P_{read} I_{read} N_{bit} \quad (8)$$

Where,  $N_b$  is number of bits accessed and  $I_{read}$  is cell current during READ. During WRITE cycle the number of bits that are being actively driven is number of bits divided by  $Y$ . The probability that those cells that are written into will be flipped is assumed to be 50%. Those cells that were not accessed for WRITE will be in the READ mode (in case where write is done through a  $Y$  to 1 muxes). Therefore, the WRITE current in the array is:

$$I_w = \left( \frac{N_b}{Y} \right) (0.5I_{flip} + 0.5I_{same}) + \left( N_b - \frac{N_b}{Y} \right) I_{read} \quad (9)$$

Where,  $Y$  is WRITE mux ratio,  $I_{flip}$  is cell current when switching state, and  $I_{same}$  is cell current when not switching state. All the cells in the array that reside in other rows may consume static power. This number can be significant when 4 transistor (4T) cells are in use. In order to determine the total leakage current, simulation must be run. This number should be multiplied by the number of cell in the entire array.

## III. LOGIC CIRCUIT

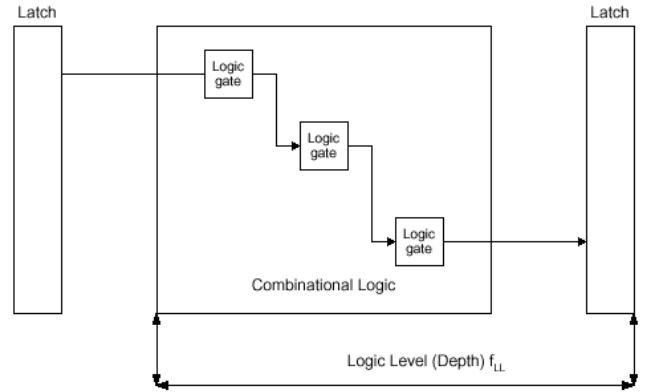


Fig. 2. A Cell built by a latch and logic gates

The random logic circuit with logic level of  $f_{LL}$  is shown in Fig 2. The logic circuit includes combinational logic followed by a latch. The power consumption from the latch is divided into  $f_{LL}$  pieces and each piece is added to one logic gate in the logic circuit. The power consumption estimation of logic circuit is divided into two parts. The first part is the power consumption of unlocked driven nodes and the second one is the power consumption of clocked driven nodes, which will be calculated in the power estimation of the clock distribution. The reason it is separated is that the switching activity factor is different between clocked driven nodes and unlocked driven nodes. Therefore, the logic circuit power consumption excluding clocking is :

$$P_{logic-circuit} = \left( \frac{1}{2} \right) A C_{total} f_{max} V_{dd}^2 \quad (10)$$

Where,  $C_{total}$  is total capacitance of logic circuit ( $C_{total} = C_{logic} + C_{latch}$ ),  $C_{logic}$  is the logic gate capacitance excluding clock driven nodes, and  $C_{latch}$  is the unlocked capacitance in a latch.  $A$  is switching activity factor.

#### IV. INTERCONNECTION

The interconnections in CMOS VLSI are divided into two different parts, which are the local and intermediate interconnections and global buses. The local and intermediate interconnections are defined as interconnections within a logic gate, between gates, or sub systems. The global bus includes data, control, and address buses.

##### A. Local and Intermediate Interconnections

$$P_{local-inter} = \left(\frac{1}{2}\right)f_{max}C_{avg-inter}V_{dd}^2 \quad (11)$$

Where,  $C_{avg-inter}$  is the total average interconnection capacitances. According to [4], the total average interconnection capacitances can be calculated as shown below:

$$C_{avg-inter} = N_g L_{avg} c_{int} \quad (12)$$

Where,  $c_{int}$ : Unit wire capacitance per unit length with a minimum wire width of  $W_{int}$ .  $N_g$  is the number of logic gates in the block.  $L_{avg}$  is the average interconnection length of a gate with a average fan in and fan out,  $f_{an_{in-out-gate}}$ .  $L_{avg} = f_{an_{in-out-gate}} l_{avg-gate}$  where  $l_{avg-gate} = \bar{R} d_{gate}$  is local and intermediate interconnection length of a logic gate.  $\bar{R}$  is average chip interconnection length in units of gate pitch.  $d_{gate}$  is the average gate dimension.

##### B. Global Bus

The bus consumes power from three different parts that are the power consumption from the bus wire capacitance, the power from the bus loads, and the power consumed in the bus drivers. Therefore, the total power consumption of the global bus is:

$$P_{global-bus} = \frac{1}{2}(C_{wire} + C_{load} + C_{driver})f_{max}V_{dd}^2 \quad (13)$$

Where,  $C_{wire}$  is the total bus wire capacitance.  $C_{load}$  is the total bus load capacitance.  $C_{driver}$  is the total capacitance of bus drivers.

#### V. CLOCK DISTRIBUTION

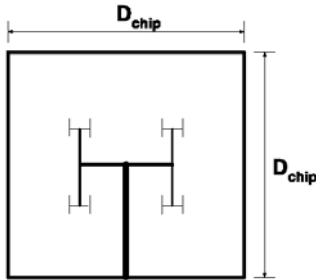


Fig. 3. Hierarchical Clock Distribution (The H-Tree Model)

Different systems may have different clock distributions. The Fig 3 above shows a H-Tree clock distribution model, which has low clock skew according to [4] and [5]. The

H-Tree clock model and the clock driver are matched at the source end. Therefore, the width of a clock wire is half its coming width before the branching points. To avoid reflections at the branching points, the line separates into two branches with characteristic impedance twice the impedance of the incoming line. In parallel, they act like a single line with the same impedance as the incoming line. If the far end clock wire has minimum width and the clock levels is 5, then the global clock wire capacitance is:

$$\left(\frac{16}{2}D_c + \frac{1 \times 8}{2}D_c + \frac{2 \times 4}{2}D_c + \frac{4 \times 2}{2}D_c + \frac{8 \times 1}{2}D_c\right) = 24D_c c_{int} \quad (14)$$

Where,  $D_c$  is chip dimension. The following equation shows the global clocking load includes all clocked transistors, all clocking wire loads, and all clock drivers:

$$C_{total-clock} = C_{clk-loads} + C_{clk-drivers} \quad (15)$$

Hence, the total power consumption from clock distribution is:

$$P_{clock} = C_{total-clock} f_{max} V_{dd}^2 \quad (16)$$

#### VI. OFF CHIP DRIVING

Off chip driving power is consumed in two parts. One is the power used to drive off chip capacitance, bonding wires, and the pad capacitance. The other part is the power consumed by the driver itself, an inverter driving chain. The first part is not given by the silicon chip technology. It is determined by the package technology and printed circuit or multichip technology. According to [6], three kinds of off chip technologies are defined. The first one is traditional package with traditional printed circuit board (PCB). The second one is advanced package and advanced PCB. The third is multichip module technology. The power consumption is:

$$P = \frac{1}{2}(C_{off-chip} + C_{off-chip-drivers})f_{max}V_{dd}^2 \quad (17)$$

Where,  $C_{off-chip}$  is the total off-chip capacitance and  $C_{off-chip-drivers}$  is the total inverter chain capacitance.

##### A. CMOS Buffer Sizing

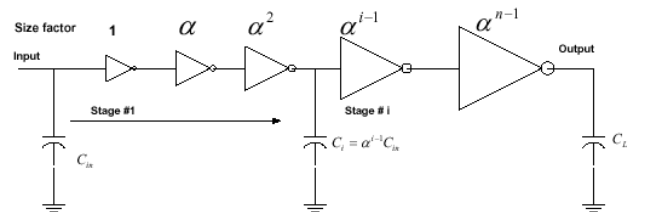


Fig. 4. Buffer Chain

When the gate is intended to drive a large load capacitance (larger than the input capacitance of the gate), the driving capability is limited and the delay is large. If we

increase the size of the gate (driver configuration), we improve the rise/fall times but still the delay can be improved by putting several stages of buffering between the first gate and the load. The objective in a buffer configuration is to get the input signal to the load as quickly as possible. Each stage in the buffer chain should have its transistor widths larger than the previous one by a factor  $\alpha$ . This is illustrated in Fig 4. According to [7], the power consumed by this inverter chain is:

$$P_{buffer} = \sum_{i=1}^n P_i = fV_{dd}^2(\alpha C_{in} + C_p) \sum_{i=1}^n \alpha^{i-1} \quad (18)$$

Hence,

$$P_{buffer} = fV_{dd}^2(\alpha C_{in} + C_p) \frac{\alpha^n - 1}{\alpha - 1} \quad (19)$$

Where,  $\alpha$  is transistor width size factor,  $n$  is the number of stages,  $C_{in}$  is input capacitance,  $C_p$  is the parasitic output capacitance, and  $C_L$  is the load capacitance.

### B. Dynamic Power Dissipation

The dynamic power dissipation of the input pad is mainly internal power. The total dynamic power of all the input pads is:

$$P_{dynamic} = AN_{inputs}E_{if} \quad (20)$$

Where,  $N_{inputs}$  is the number of the input pads and  $E_{if}$  is the internal energy of the input pad in Watt/Hz.

## VII. GATE COUNT ESTIMATION FROM VERILOG RTL DESCRIPTIONS

A procedure to estimate the gate count of Verilog RTL descriptions is presented in this section. Although this method uses many simplifications to calculate the gate count of a circuit, results are fast and accurate enough to help designers to improve their descriptions. The gate count estimation tool, which is implemented in C language, was developed and the way it works is shown in Fig 5. Basically, the gate count estimation tool takes a Verilog design description file as an input and it estimates the number of gates needed to implement the design description as an output. Estimation are based on generic library, so their results are given in terms of a reduced set of primitives. Every synthesizable Verilog statement is estimated considering gates like AND, NAND, OR, NOR, XOR, XNOR, inverters and multiplexer for the combinational logic. Apart from the inverter, all gates have two inputs and one output. In order to estimate the gate-count of each statement, they are divided in smaller parts, logic or arithmetic expressions being the smallest parts. Every single part is calculated independently, and all single results are added into the global statement gate-count result.

### Example: Code 1

If we have a Verilog conditional signal assignment as shown below:

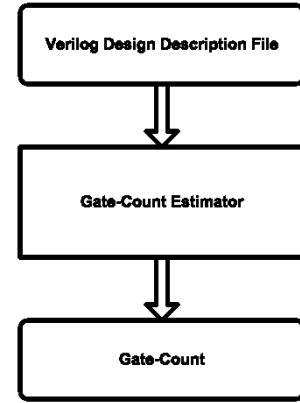


Fig. 5. Gate-Count Flowchart

```

initial
begin
  if (C > D)
    S = A & B; // A and B
  elseif (C == D)
    S = ~(A | B); // A nor B
  else
    S = A ^ B; // A xor B
  
```

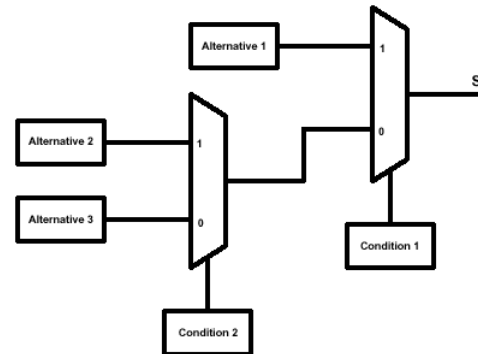


Fig. 6. Hardware Generated for Code 1

- The statement in Code 1 is composed of:
- Bitwise operators in waveforms (A and B, A nor B, and A xor B)
  - Relational operators in conditions (C < D and C = D)
  - The blocking assignment and conditional statements (=...if...else...)

To estimate the final gate-count, every expression is computed independently. In this case, every **if** in the statement generates a multiplexer [9], [12], and [13]. To simplify the example, it is considered that signals A, B, C, D are one bit width, but considering other widths only implies to multiply every local result by the width of the signals involved. Fig 6 shows the generated hardware. The conditional statement has generated two multiplexers as it can be seen in Fig 6 above. The conditions and alternatives are computed separately and produce the following hardware as shown in Fig 7. In any case, the structure generated by

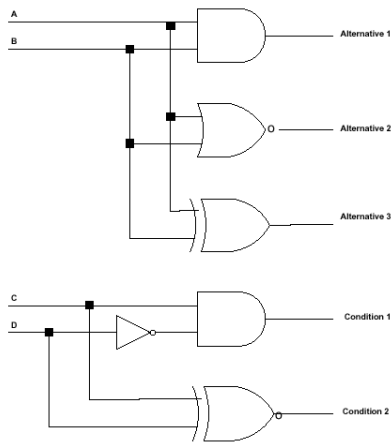


Fig. 7. Expressions and Conditions of Code 1

the conditional statement is always the same. Adding the gate count of each part, the final gate count for the code 1 above is obtained.

$$gate - count_{total} = N_{muxes} + N_{alternatives} + N_{conditions} \quad (21)$$

## VIII. CONCLUSION

A method for estimating power consumption of digital CMOS VLSI chips is developed. The method estimates the total power consumption of a chip by analyzing separately for different parts of the chip, which are logic circuit, on chip memory, interconnection, clock distribution, and off chip driving. This method makes it possible to estimate the power consumption of a digital CMOS VLSI chip based on gate count, memory size, and logic circuit. An estimation tool, which is implemented in C language, is created and used to estimate the gate count from Verilog register transfer level (RTL) descriptions.

## REFERENCES

- [1] W. H. Lien and W. P. Burlison, "Wave-Domino Logic: Theory and Applications", Proceedings of TAU'92, 1992.
- [2] M. Nemani and F. N. Najm, "Towards a high-level power estimation capability", IEEE Transactions on Computer-Aided Design, vol. 15, no. 6, pp. 588-598, June 1996
- [3] T. M. Cover and J. A. Thomas, "Elements of Information Theory", New York, NY: John Wiley & Sons, 1991
- [4] H. B. Bakoglu, "Circuits, Interconnections, and Packaging for VLSI", (VLSI systems series). Addison-Wesley Publishing Co., Reading, Mass., 1990
- [5] S. H. Unger and C. Tan, "Clocking schemes for high speed digital systems", IEEE Transactions on Computer-Aided Design, vol. 35, pp. 880-895, 1986
- [6] S. K. Tewksbury, "Wafer-level integrated systems: Implementation issues", (VLSI Computer Architecture and Digital Signal Processing Series). Reading, MA: Kluwer Academic, 1989
- [7] A. Bellaouar and M. I. Elmasry, "Low-Power Digital VLSI Design Circuits and Systems", Kluwer Academic Publishers, 1995
- [8] Wayne Wolf, "Modern VLSI Design Systems on Silicon", Second Edition, Prentice-Hall, 1998
- [9] F. Machado, Y. Torroja, F. Casado, T. Riesgo, E. de la Torre, and J. Uceda, "A simple method to estimate the area of VHDL RTL descriptions", Design of Circuits and Integrated Systems (DCIS) Montpellier (France), november 2000
- [10] Samir Palnitkar, "Verilog HDL A Guide to Digital Design and Synthesis", Prentice-Hall, 1996

- [11] Zainalabedin Navabi, "Verilog Digital System Design", McGraw-Hill, 1999
- [12] Roger Lipsett, Carl Schaefer, and Cary Ussery, "VHDL: Hardware Description and Design", Kluwer Academic Publishers, 1989
- [13] James R. Armstrong and F. Gail Gray, "Structured Logic Design with VHDL", Prentice-Hall, 1993
- [14] Donald E. Thomas and Philip R. Moorby, "The Verilog Hardware Description Language", Second Edition, Kluwer Academic Publishers, 1995