



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Voltage and Frequency Scaling in Low Power VLSI

DR. K.Padmapriya^{*1}, CH.Pavankumar²

^{*1,2}ECE Department JntukceV. Vizianagaram, India

kesaripadmapriya@yahoo.com

Abstract

In micro electronics design, power consumption, speed of operation, are crucial constraints. Propagation delay of circuit component has an impact on such factors. This paper investigates the effect of supply and threshold voltages and frequency at which the VLSI chip is operated and the desired techniques for lowering the voltage and frequencies to obtain the low power consumed VLSI system. Some special techniques which can reduce the clock frequency like pipelining and parallel processing strategies for desirable propagation delays are explained in brief in this paper

Keywords: voltage scaling, frequency scaling, power reduction in VLSI.

Introduction

The increasing prominence of portable systems, like notebook computers, portable communication devices demanding high chip density and high throughput along with low power consumption. Now a days reducing the power consumption has become an important objective in the design of digital integrated circuits. This can be done through design improvements.

The total power consumption is divided among four major parts.

- 1.Logic circuits
- 2.Clock generation and distribution
- 3.Interconnections and
- 4.Off-chip driving (I/O circuits).

The two main sources of power dissipation in VLSI circuits are static power and dynamic power. Static power [1] results from resistive paths between power supply and ground. The static power dissipation of a circuit is expressed by the relation

$$P_{stat} = I_{stat}V_{DD} \text{-----(1)}$$

Where,

I_{stat} is the current that flows between the supply rails in the absence of switching activity

Dynamic power results from switching capacitive loads between different voltage levels.

For a CMOS gate the dynamic power is

$$P_{dynamic} = \alpha CV_{DD}^2 \frac{1}{T} \text{----- (2)}$$

Where,

α is the activity factor of the output nodes.

C is the total capacitance of the output nodes.

V is the supply voltage.

T is the delay or time period.

For a complex chip, the total dynamic power is the sum of the dynamic power of all the gates. Resultant equation is same as equation (2). The only difference is which is the total capacitance of all the loads and the activity factor, which is the average activity factor. So, equation (2) is modified as

$$P_{dynamic} = \alpha C_{eff} V_{DD}^2 \frac{1}{T} \text{----- (2.1)}$$

By summing up these two dissipations, total power dissipation is not obtained. Still there is some difference which is caused by two other factors i.e., due to the short circuit current and the leakage current. Short circuit current results from both the transistors in a CMOS inverter being ON at the same time when the input switches and there are four main sources of leakage current in CMOS transistor [2]. They are

1. Reverse-biased junction leakage current (I_{REV})
2. Gate induced drain leakage (I_{GIDL})
3. Gate direct-tunnelling leakage (I_G)
4. Sub threshold (weak inversion) leakage (I_{SUB})

Thus, the total power dissipation in CMOS digital circuits can be expressed as the sum of four components [3]

$$P_{total} = \alpha C_{load} V_{DD}^2 / \tau + V_{DD} (I_{(short-circuit)} + I_{leakage} + I_{static}) \quad (3)$$

Where,

$I_{short-circuit}$ = average short circuit.

$I_{leakage}$ = reverse leakage and sub threshold current.

I_{static} = DC current component drawn from the power supply

The first term in the equation (3) is the dominating component in most CMOS logic circuits (which is due to dynamic power) because it is proportional to the switching frequency, which we are increasing rapidly to improve the speed of the circuit. The effect of the remaining terms on the circuit is very less. So, they can be ignored. [4][5]

Device characteristics (e.g., threshold voltage), device geometrics, and interconnect properties are significant factors in lowering the power consumption. Circuit-level measures such as the reduction of voltage swing, clocking strategies can be used to reduce power dissipation at the transistor level. The power consumed by the system can be reduced by minimizing the number of switching events for a given task.

We have several means for reducing the power consumption.

1. Reduction of the power supply voltage.
2. Reduction of voltage swing in all nodes.
3. Reduction of the switching probability (transition factor).
4. Reduction of the load capacitance.

Switching power dissipation is also a linear function of the clock frequency .But reducing it diminishing the overall system performance. Thus, the reduction of clock frequency would be a viable option only in cases where the overall throughput of the system can be maintained by other means.

The reduction of switching activity requires a detailed analysis of signal transition probabilities, and implementation of various circuit-level and system level measures such as logic optimization, use of gated clock signals, and prevention of glitches.

The load capacitance can be reduced by using certain circuit design styles and by proper transistor sizing.

Voltage and Frequency Scaling[6]

Reduction of supply voltage, V_{DD} is nothing but voltage scaling. As the dynamic power is proportional to the square of the operating voltage, reducing it significantly improves the power consumption. Furthermore, frequency is directly proportional to supply voltage from equation (2) .

Relation between voltage and frequency is as shown in figure 1 [7]

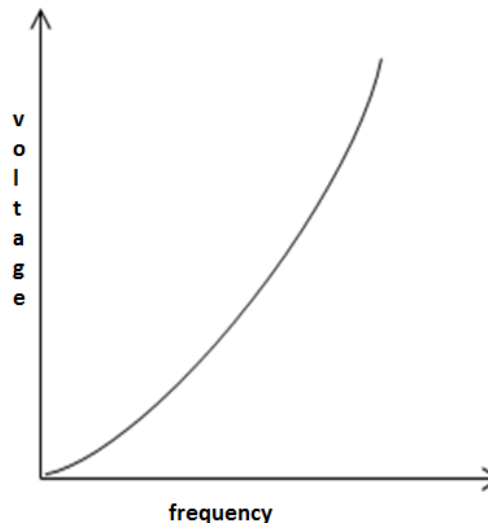


Figure 1 relation between frequency and voltage

This dependency of frequency on supply voltage scales frequency and it can be lowered. Therefore, a cubic power reduction is possible. Here, assumption is that switching frequency and load capacitance are constant because we are dealing with only supply voltage.

Influence of Voltage Scaling on Power And Delay[3]

Although the reduction of power supply voltage significantly reduces the dynamic power consumption, the inevitable design trade-off is the increase of circuit delay and decrease of clock speed. This can be seen by the propagation delay expressions for the CMOS inverter circuit.

$$\tau_{PHL} = \frac{C_{load}}{k_n(V_{DD} - |V_{T,n}|)} \left[\frac{2|V_{T,n}|}{V_{DD} - |V_{T,n}|} + \ln \left(\frac{V_{DD} - |V_{T,n}|}{V_{DD}} - 1 \right) \right] \quad (4)$$

$$\tau_{PLH} = \frac{C_{load}}{k_p(V_{DD} - |V_{T,p}|)} \left[\frac{2|V_{T,p}|}{V_{DD} - |V_{T,p}|} + \ln \left(\frac{V_{DD} - |V_{T,p}|}{V_{DD}} - 1 \right) \right] \quad (5)$$

The delay of circuit depends on power supply voltage as [9]

$$\tau = k C_L V_{DD} / (V_{DD} - V_t)^2 \quad (6)$$

Where τ is the circuit delay

k is the gain factor

C_L is the load capacitance

V_{DD} is the supply voltage
 V_t is the threshold voltage

Thus, by reducing the voltage, although we can achieve cubic power reduction, the execution time increases. The main challenge in achieving power reduction through voltage and frequency scaling is therefore to obtain power reduction while meeting all the time constraints.

To reduce the effect of power supply voltage upon delay, the threshold voltage of the transistor (V_T) is scaled down accordingly. However, this approach is limited because the threshold voltage may not be scaled to the same extent as the supply voltage. When scaled linearly, reduced threshold voltages allow the circuit to produce the same speed - performance at a lower V_{DD} . But smaller threshold voltages lead to smaller noise margins for the CMOS logic gates. The sub threshold conduction current also sets a severe limitation against reducing the threshold voltage. For threshold voltages less than 0.2V, leakage due to sub threshold conduction in stand-by i.e., when the gate is not switching, may become a very significant component of the overall power consumption. In practice, extending the voltage range below half V_{DD} is effective, but extending this range to sub-threshold operations may not be beneficial.

Techniques to Overcome Difficulties Associated With Low- V_t Circuits

There are two circuit design techniques which can be used to overcome the difficulties associated with the low- V_T circuits. They are

1. Variable-Threshold CMOS (VTCMOS) and
2. Multiple-Threshold CMOS (MTCMOS).

Variable-Threshold CMOS (VTCMOS) Circuits[3][8][9]

We have seen that using a low supply voltage (V_{DD}) and a low threshold voltage (V_T) in CMOS logic circuits is an efficient method for reducing the overall power dissipation, while maintaining high-speed performance. But there are some difficulties with low- V_T transistors. One possible way to overcome this problem is to adjust the threshold voltages of the transistors in order to avoid leakage in the stand-by mode, by changing the substrate bias.

As the threshold voltage V_T of an MOS transistor is a function of its source-to-substrate voltage V_{SB} , connecting substrate terminals of all nMOS transistors to ground potential and substrate terminals of all pMOS transistors to V_{DD} ensures that the source and drain diffusion regions always remain

reverse-biased with respect to the substrate, and the threshold voltages of the transistors are not significantly influenced by the body effect.

The VTCMOS technique can also be used to automatically control the threshold voltages of the transistors in order to reduce leakage currents, and to compensate for process-related fluctuations of the threshold voltages. This approach is also called the Self-Adjusting Threshold-Voltage Scheme.

This technique is very effective for reducing the sub threshold leakage currents and for controlling threshold voltage values in low V_{DD} -low V_T applications. However, this technique requires twin well or triple-well CMOS technology in order to apply different substrate bias voltages to different parts of the chip. Also, separate power pins may be required if the substrate bias voltage levels are not generated on-chip. The additional area occupied by the substrate bias control circuitry is usually negligible compared to the overall chip area.

Multiple-Threshold CMOS (MTCMOS) Circuits[3][10][11]

This is the technique used to reduce leakage currents in low-voltage circuits in the stand-by mode based on using two different types of transistors (both nMOS and pMOS) with two different threshold voltages in the circuit.

Here, low $-V_T$ transistors are typically used to design the logic gates where switching speed is essential, whereas high- V_T transistors are used to effectively isolate the logic gates in stand-by and to prevent leakage dissipation.

The MTCMOS technique is conceptually easier to apply and to use compared to the VTCMOS technique, which usually requires a sophisticated substrate bias control mechanism. It does not require a twin-well or triple-well CMOS process. But, there is a difficulty in this is that the fabrication of MOS transistors with different threshold voltages on the same chip. One more disadvantage is that the MTCMOS circuit technique consists series - connected stand-by transistors, which increase the overall circuit area and also add extra parasitic capacitance and delay.

While these techniques can be very effective in designing low power VLSI circuits, they may not be used as universal solutions. Because in certain applications, variable threshold voltages and multiple threshold voltages have technical limitations. So, pipelining and hardware replication techniques became alternatives for maintaining system performance despite voltage scaling.

Some of the scaling techniques to reduce power dissipation are explained below.

Design-Time Voltage and Frequency Setting[6]

This is the most common technique to reduce power consumption i.e., by scaling the voltage during design time. Design time schemes scale and set the voltage and frequency, which remains constant for all applications at all time .So, it is efficient and few more techniques are introduced.

Static Voltage and Frequency Scaling[6]

In static voltage and frequency scaling, systems can be designed for several voltage and frequency levels, which can be switched at run time. The change to a different voltage and frequency is pre-determined .But, there are many difficulties in this scaling procedure also.

Timing analysis for multiple designs is complicated as the analysis has to be carried out for different voltages .This requires libraries characterized for the different voltages used. Constraints are specified for each supply voltage level or operating point .There can be different operating modes for different voltages. Constraints need not be same for all modes and voltages. The performance target for each mode can vary. Timing analysis should be carried out for all these situations simultaneously. Different constraints at different modes and voltages have to be satisfied.

Dynamic Voltage and Frequency Scaling[6][7]

These techniques are implemented for energy management of real time systems. The application and system characteristics can be dynamically analyzed to determine the voltage and frequency settings during execution. Devices dynamically change their speed increasing the energy operation efficiency. The reduction of energy consumption in systems can be achieved without affecting the performance. These techniques are able to make energy savings while providing the necessary peak computation power in VLSI based systems[12][13][14].An example of dynamic voltage scaling technique is Adaptive voltage scaling technique[15] which make the decision on the next setting based on the recent history of execution.

Frequency Scaling Techniques

As the power is directly proportional to the clock frequency f_{clk} , we can reduce the power dissipation by using different techniques of frequency scaling.

For high performance system design propagation delay minimization plays an important role. To investigate and analyse data flow and data paths i.e., parallelism and pipelining among tasks and sub tasks , system modelling methods like block diagrams, signal flow graph(SFG), data flow graph(DFG), dependence graph etc..., are very much required. In such designs there is a trade off between sampling frequency, operating frequency and power consumption in order to design high performance systems.

Various concepts such as pipelining, parallel processing, retiming, unfolding, systolic array etc..., are used in design of modern VLSI based low power. A system is always preferred to be with high clock speed or low power consumption. In order to transform original sequential circuit to another circuit to realize these specifications, pipelining is used.

Pipelining[6][7]

Pipelining is used to transform a sequential Circuit to that circuit which has high clock speed or sample speed or low power consumption. It reduces the critical path which will increase sample speed as well as clock speed, thus speed of operation is improved. Critical path is the longest computation path among all paths and its computation time is taken as the lower bound on the clock period of the circuit. For example consider a sequential circuit with 6-Tap FIR filter. It consists of multipliers and adders as shown in figure 2.Computation time for adder and multiplier is 10u.t. & 14u.t. respectively. The critical for this circuit is 64u.t. Minimum clock period required for the execution is 64u.t.Sampling time is 64u.t. and hence sampling frequency is 64u.t. In order to increase clock frequency as well as sampling frequency pipelining is used in figure 3. In this Pipelining, latches are placed which reduces the critical to 24u.t. is shown in figure 4. Hence, reduces sample and clock duration but increases circuit complexity, latency and power consumption.

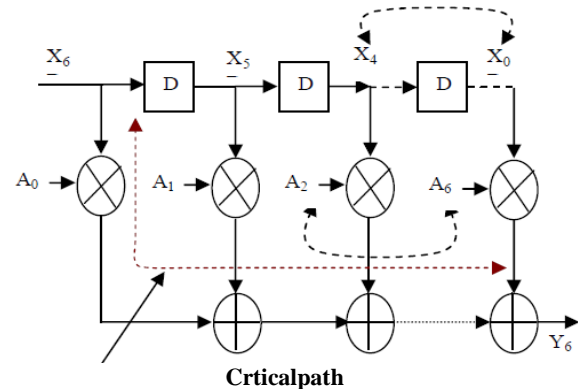


Fig2.blockdiagram of 6-tap fir filter

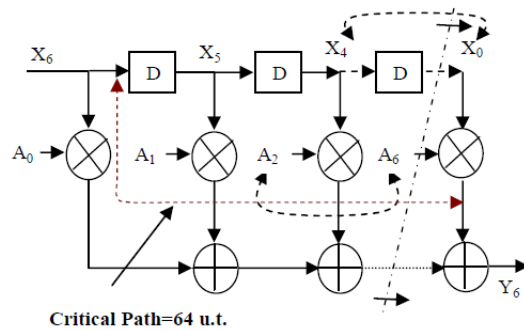


Fig 3. block diagram of 6-tap fir filter with feed forward cutest (pipelining)

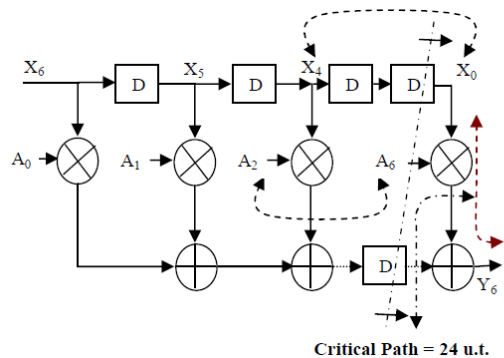


Fig 4. block diagram of 6-tap fir filter with feed forward cutest and delays (pipelining)

Level of pipelining deals with the number of latches connected between the nodes. In pipelining sample rate (Sr) is equal to clock period (Tc)

$$Y_{(n)} = \sum(A_i x_{n-i}) \dots \dots \dots (7)$$

Where,

- A_i is arbitrary values
- x_{n-i} is input signal
- $Y_{(n)}$ is out put signal
- $i = 0,1,2,3 \dots (N-1)$ N is integer, n is number of Tap FIR filters

For 6 Tap FIR filters o/p is given by

$$Y_6 = A_0 X_6 + A_1 X_5 + A_2 X_4 + A_3 X_3 + A_4 X_2 + A_5 X_1 + A_6 X_0 \dots \dots (8)$$

Consider another functional block in figure 5, which implements a logic function F (INPUT) of the input vector, INPUT. Input & output vectors are sampled through register arrays driven by a clock signal CLK. Assume that the critical path in this logic block is t_{clk} , The maximum propagation delay

between input & output is less than or equal to $T_{clk} = 1/f_{clk}$.

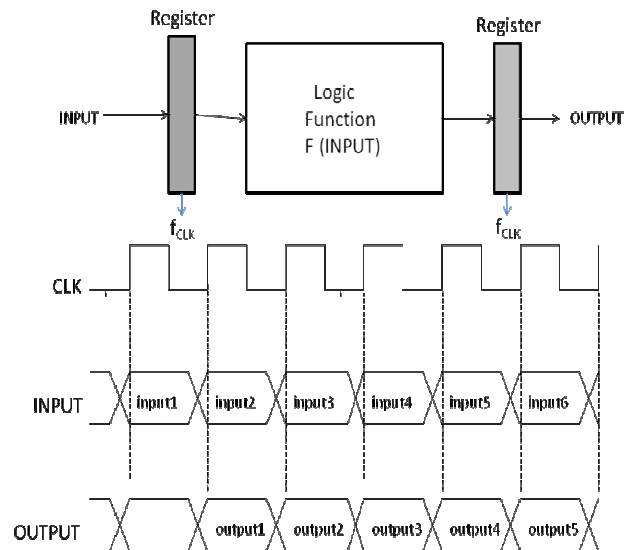


Figure 5. Single stage implementation of a logic function and its simplified timing diagram.

A new input vector is latched into the input register array at each clock cycle and output data is valid with a latency of one clock cycle. Let C_{total} be the total capacitance switched for every clock cycle. Here C_{total} consists of

- I. Capacitance switched in the input register array
- II. Capacitance switched to implement logic
- III. Capacitance switched in the output register array

Dynamic power consumption of this block is

$$P_{reference} = C_{total} V_{DD}^2 f_{CLK} \dots \dots (9)$$

Now consider N stage pipelining structure in figure 6 to implement the same logic function.

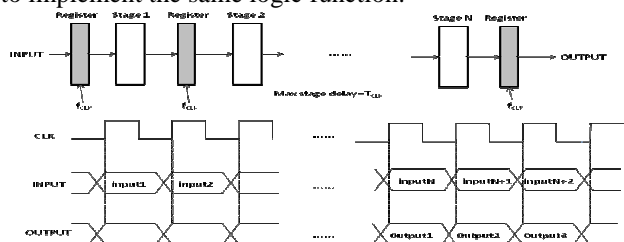


Figure 6. N-stage pipeline structure realizing the same logic function as in fig 5. The maximum pipeline stage delay is equal to the clock period, and the latency is N clock cycles.

The logic function F(INPUT) has been partitioned into N successive stages, and (N-1) register arrays are added in addition to input and output registers. All these registers are clocked with the original sample rate. The delay in this structure is

$$\tau_p \text{ (pipelining stage)} = T_{CLK} \dots (10)$$

From the above equation, we can say that the logic blocks can be operated N times slower than the original one but with same throughput. So the power supply voltage can be reduced to a value of

The dynamic power consumption of the N-stage pipelined structure with a lower supply voltage but with same functional throughput is

$$P_{\text{pipeline}} = [C_{\text{total}} + (N - 1)C_{\text{reg}}] V_{DD, \text{new}}^2 f_{\text{CLK}} \dots (11)$$

Where C_{reg} is the capacitance switched by each pipeline register. And power reduction factor in a N-stage pipeline structure is

$$\frac{P_{\text{pipeline}}}{P_{\text{reference}}} = \frac{[C_{\text{total}} + (N - 1)C_{\text{reg}}] V_{DD, \text{new}}^2 f_{\text{CLK}}}{C_{\text{total}} V_{DD}^2 f_{\text{CLK}}} = \left[1 + \frac{C_{\text{reg}}}{C_{\text{total}}} (N - 1) \right] \frac{V_{DD, \text{new}}^2}{V_{DD}^2} \dots (12)$$

For example, consider a single stage logic block with $V_{DD}=5\text{v}$ and $f_{\text{CLK}}=20\text{MHZ}$ is replaced by a four stage pipeline structure running at the same clock frequency. This means that the propagation delay of each block can be increased by 4 times. Assume that the magnitude of the threshold voltage of all the transistors is 0.8V , to acquire the required speed reduction supply voltage is to be reduced from 5V to 2V . This means by introducing pipelining into a sequential circuit 80% of the power is saved.

The architectural modification requires relatively a small area overhead. A total of (N-1) registers arrays have to be added, while trading off area for lower power latency is increased from 1 to N clock cycles.

Parallel Processing[6][7]

Parallelism is a method for trade off area for lower power dissipation. This approach is useful especially when the logic function to be implemented is not suitable for pipelining. In parallel processing, multiple input samples can be processed for the same clock pulse. In this sampling time T_{sample} is not equal to the clock duration T_{clock} . To increase sampling rate for the same clock time, number of sequential

hardware can be connected in parallel as shown in figure 7. In this diagram sequential hardware consisting 6-Tap filter as a nodes A and B are connected in parallel. $X(Lk)$ to $X(Lk+m)$ samples from input signal are processed in single clock pulse with duration T_c , to get output samples, where $m=L-1$. Single input single output (SISO) system must be converted into multiple input multiple output (MIMO) system.

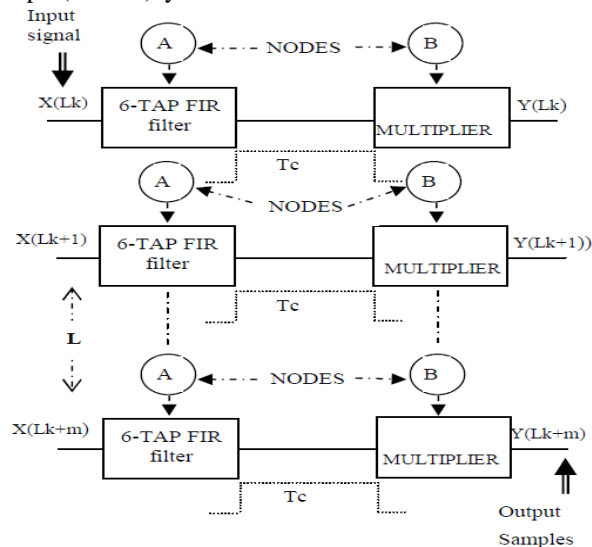


Fig 7. level of parallel multiple input multiple output (mimo) system

Level of parallelism (L) depends on the number of sequential circuits connected in parallel. It supports increase in sample rate, to increase speed of architecture thereby processing number of samples in a single clock pulse. Initially critical path will not change but fine grain pipelining can be used for further reduction in critical path. In fine grain pipelining, multipliers are broken into sub multipliers of different computation time, which will reduce critical path in parallel architecture.

Another way of explaining parallel processing is that consider N identical processing elements, each implementing the logic function F (INPUT) in parallel, as shown in figure 8.

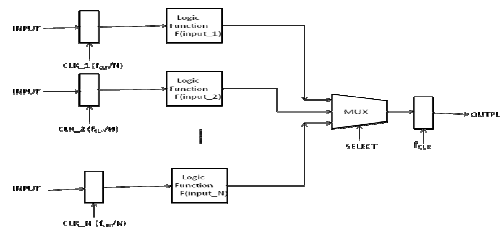


Figure 8. N-block parallel structure realizing the same logic function as in fig 5. Notice that the input registers are clocked at a lower frequency of (f_{CLK}/N)

Assume that the consecutive input vectors arrive at the same rate as in the single stage. The input vectors are routed to all the registers of the N processing blocks. Gated clock signals, each with a clock period of (NT_{clk}), are used to load each register every N clock cycles. This means that the clock signals to each input register are skewed by T_{clk} , such that each one of the N consecutive input vectors is loaded into a different input register. Since each input register is clocked at a lower frequency of (f_{clk}/N), the time allowed to compute the function for each input vector is increased by a factor of N. This implies that the power supply voltage can be reduced until the critical path delay equals the new clock period of (NT_{clk}). The output of the N processing blocks are multiplexed and sent to an output register which operates at a clock frequency of f_{clk} , ensuring the same data throughput rate as before. The timing diagram of this parallel arrangement is shown in fig 9

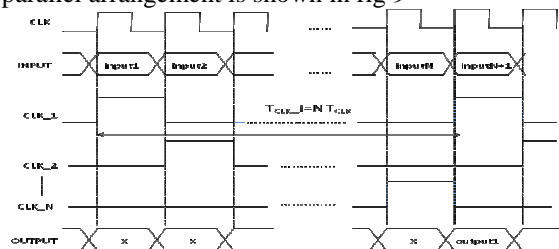


Figure 9. Simplified timing diagram of the N-block parallel structure

Since, the time allowed to compute the function for each input vector is increased by a factor of N, the power supply voltage can be reduced to a value of $V_{DD,new}$, to effectively slowdown the circuit. The total dynamic power dissipation of the parallel structure is

$$P_{parallel} = NC_{total}V_{DD,new}^2 \frac{f_{CLK}}{N} + C_{reg}V_{DD}^2 f_{CLK}$$

$$= (1 + \frac{C_{reg}}{C_{total}})C_{tot} \dots \dots \dots (13)$$

There is also additional over head. If it is neglected, the amount of power reduction achievable in a N-block parallel implementation is

$$\frac{P_{parallel}}{P_{reference}} = \frac{V_{DD,new}^2}{V_{DD}^2} \dots \dots \dots (14)$$

The lower bound of switching power reduction realizable with architecture driven voltage scaling is found, assuming zero threshold voltage as

$$\frac{P_{parallel}}{P_{reference}} \approx \frac{1}{N^2} \dots \dots \dots (15)$$

The Consequences of this processing are increased area overhead and latency.

In case of communication bounded system, where it is not possible to reduce critical path further, parallel and pipelining can be combined to increase the speed of the architecture.

Conclusion

In this paper significance of power consumption, voltage and frequency scaling techniques such as static scaling, dynamic scaling to reduce power consumption is explained. VTCMOS, MTCMOS techniques are explained which reduces the difficulties associated with low threshold voltage. Speed of operation with reference to pipelining and parallel processing VLSI techniques are analyzed. It is observed that parallel processing, pipelining and voltage scaling can reduce the power consumption.

References

- [1] Jan M.Rabaey , Anantha Chandra kasan and Borivoje Nikolic , “Digital integrated circuits, a design perspective ,second edition”.
- [2] Farzan Fallah and Massoud Pedram, “standby and active leakage current control and minimization in CMOS VLSI circuits”.
- [3] Sung-mo kang and leblibici, “cmos digital integrated circuits”.
- [4] A. Chatterjee, M. Nandakumar, and I. Chen, “An investigation of the impact of technology scaling on power wasted as short current in low voltage CMOS,” in *IEEE Int. Symp. Low Power Electronics and Design*, Aug. 1996, pp. 145–150.
- [5] Bavier, A., B. Montz, and L. Perterson, 1998. Predicting MPEG Execution Times, SIGMETRICS/PERFORMANCE '98, Int'l Conf. on Measurement and Modelling of Computer Systems, pp: 131-140
- [6] Massoud Pedram, Department of EE-Systems, University of Southern California, “basic low power digital design”.
- [7] A significance of VLSI techniques for low power real time systems.
- [8] Isbal Y. Yang, Carlin Vieri, Anantha Chandrakasan, Dimitri A. Antoniadis, “Back gated CMOS on SOIAS for Dynamic Threshold Voltage Control”.

- [9] Fariborz Assaderaghi, Stephen Parke, Member IEEE, Dennis Sinitsky, Jeffrey Bokor, Member, IEEE, Ping K. Ko, Senior Member IEEE, and Chenming Hu, Fellow, IEEE, "A Dynamic Threshold Voltage MOSFET (DTMOS) for very low voltage operation".
- [10] Shin'ichiro Mutoh, member, IEEE, Takakuni Douseki, member, IEEE, "1-V Power Supply High-speed digital circuit technology with Multithreshold-voltage CMOS".
- [11] Takakuni douseki, Santoshi Shigematsu, Yasuyuki Tanabe, Mitsuru Harada, Hiroshi Inokawa, and Toshiaki Tsuchiya. "A 0.5V SIMOX-MTCMOS Circuit with 200ps Logic Gate".
- [12] Flavius Gruian, "Hard real time scheduling for low energy using stochastic data and DVS processors," in proc. of Int. symposium on Low Power Electronics and Design California, USA, 2001, 46-51.
- [13] Youngsoo Shin, Kiyong Choi, Takayasu Sakurai, "Power optimization of real time embedded systems on variable speed processors," in proc. of IEEE ACM international Conference on Computer Aided Design, 2000, 365-368.
- [14] C.M. Krishna, Yann-Hang Lee, "Voltage clock scaling adaptive scheduling techniques for low power in hard real-time systems," IEEE transaction, Computers, Vol.52, No. 12, Dec.2003, 1586-1593.
- [15] Mohamed Elgebaly and Manoj Sachdev, "Efficient Adaptive scaling system through on-chip critical path emulation".