# Power Efficient Processor Design Techniques and Approaches

Himanshu Dinesh Chhabra
Department of Electrical Engineering and Computer Science
Syracue University
Syracuse USA

hdchhabr@syr.edu

Ankita Kisan Gadage
Department of Electrical Engineering and Computer Science
Syracue University
Syracuse USA,

agadage@syr.edu

***Abstract*—In the past few decades, the demand for portable handheld mobile devices have taken a significant boom. The miniaturization of processors and feature size of the mobile technology has increased the demand for computationally powerful and energy demanding mobile devices. Continuous miniaturization in the chip technology has typically increased the power utilization. Recent research efforts in designing low powered micro-computers has resulted in several novel and alternative techniques that mainly focus on improvements at the processor component and architecture levels without compromising the throughput.**

**This paper comprehensively provides an overview of the basics of low-powered processor design fundamentals along with several power-efficient design techniques. The paper highlights the significant power demanding contributors and provides alternative power efficient techniques including DFVS, Power and Clock Gating to reduce the energy dissipation at component levels. Additionally, details on energy optimization by reducing static power leakages, techniques to optimize cache power consumption has also been discussed in this paper. Techniques to redesign energy-efficient interconnection buses and pipeline component improvement techniques like Pipeline Gating and Balancing are also discussed in this paper. Detailed Overview to achieve power optimization at the instruction issue logic level, designing of ultra-low power processors with the use of value compression technique and low-powered processor architecture is discussed.**

***Keywords—power gating, cache optimization, power efficient architectures, energy-dissipation, static and dynamic power leakage, power optimization techniques***

## I. Introduction And Background

The amount of power consumed by the processors have a direct impact on the design of the processors. Power consumption employs key constraints for embedded systems as the battery life is a critical aspect. Moreover, for high performance requirements are accompanied with high power dissipation which also have a direct impact on the costs estimates for cooling systems. Since a decade there has been a significant effort towards increasing the battery-life and inventing better ways to cool down the system but minuscule efforts towards building low-powered processors.

Since the advent of transistors technology, the main focus has always been over performance improvement and reduction of the feature size, although they require very small amount of energy and power to perform capacitive switching but the overall effect of millions of transistors on the chip have a great effect on power consumption. With the invention and ever-growing usage of portable and mobile devices, low-power architecture design has become the need of the hour in the micro-electronics field. Moreover, with increase in expectations with handheld devices such as image processing and speech recognition, more power and space is required by the processors, thus power-efficiency and novel techniques to deliver the same quality of features with low battery-power consumption is a recent research effort. In-order to incorporate more features miniaturization is necessary but the power consumed per unit area on the feature chip is exponentially increasing, given below is the image which gives a good idea about magnitude of power consumed (power-density) by commercially available intel micro-processors. It is quite clear from the picture that power-density of intel-family processors have exponentially increased.

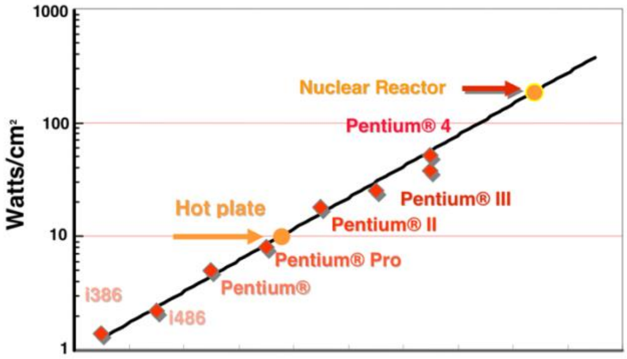


Fig1. Intel Family Processor on chip power consumption [1]

At such high-power dissipation, heat generated is quite high and present cooling mechanism either turn out to be ineffective or very expensive. In CMOS-Circuitry within processors, the major source of power dissipation is due to the resistance offered by the medium within the transistors, the magnitude of power dissipation is usually proportional to the supplied voltage. Embedded processors have also been a latest research area amining at designing power efficient and high perfromance chips. This can be used by the modern application specific processors to increase functionality. For instance, we can include fixed memory space for functional-unit as opposed to including memory on the chip. In this way bottleneck is avoided by reducing data flow in the system. Thus it should be noted that, the energy comsumption of the system can be reduced, by effective utlization of the area space using locality of reference as a tool.

As the sophistication in technology is increasing, energy utilization has become one of the crucial parameters. Thus, efforts have to be invested in achieving less power utilization at system component and designing levels. It has been discovered that the advancements in the field of low power processing, progress is mainly done at the component level:

low power display, low power CPU, low power receivers etc. There is not much system level low power optimization. Low power components area necessary for portable systems but attention must be given to lower power hardware and software architecture as well. Overall lower power system architecture is beneficial as there a inter connections between the subsystems thus enhancement of just one component increases the chances of energy utilization by other system components. Energy optimization techiques have to be applied on all system design statges as there is a close association amongst the computer hardware, applications and the operating systems. Firstly, we need to have components with latest power optimization technology, then we need to optimize power utilization at the system and levels. All these three techniques combined can reduce the power utilization by a considerable amount. Moreover, optimization of power consumption at the hardware is not enough we need to also optimize the operating system and applications. Application directly interact with the user and thus play a vital role in controlling the power utilization of the system

This paper is organized to present various energy optimization techniques that can be used of making an optimized system that uses less power. Following the introductory part the paper briefly describes the main contributors and causes of power dissipation in present computer architecture. Subsequently the sections comprehensively describe new novel alternative techniques at processor component level mainly targeting processor caches, instruction issue techniques, instruction pipelines, interconnected buses structure along with ultra-low powered techniques and others at architectural level. Finally, the last section derives conclusions based on the proposed novel techniques to design and develop low-powered machines.

## II. Analysis

### A. Overview

Several Processor parameters including voltage, cache, instruction pipeline, issue logic designs, can be improved to achieve an overall reduction in power consumption by the processor..

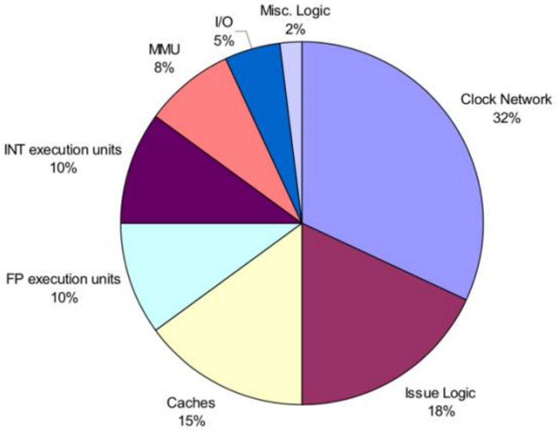
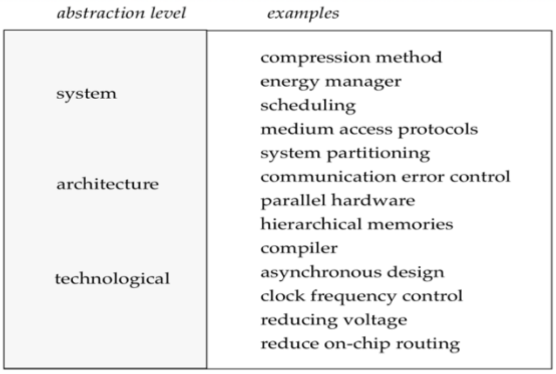


Fig 2. Power Consumption Chart [1]

The image above gives a good idea of major contributors to power consumption. The section below addresses major sectors and discuss several novel techniques to achieve a low-powered processor architecture model

### B. Characteristics of Low power design

While designing a system the main concern that designer face is to reduce energy consumption. Battries being power restricted devices with a fixed amout of energy, power consumption is. Power is energy utilized per unit time and considering that the power utilization remains unchanged the energy soley depends only on the time for which it is utilized. Decreasing power utilization will result in conserving energy only if the time needed to complete the task is less. A system that uses more power than another may not use more energy for some kind of applications. For instance, if system A needs two times the power utilization of system B then A’s energy requirement will be less if it can execute the same program in less time than that of B.for example, do not differentiate among departments of the same organization).



| abstraction level | examples |
|---|---|
| system | compression method<br>energy manager<br>scheduling<br>medium access protocols |
| architecture | system partitioning<br>communication error control<br>parallel hardware<br>hierarchical memories |
| technological | compiler<br>asynchronous design<br>clock frequency control<br>reducing voltage<br>reduce on-chip routing |

Fig 3. Optimization levels [2]

*1) Overall System Design Modifications and Implications:* The design structure of system consists of various level of optimization. While designing a power optimized system, all levels of the design flow must receive equal importance. Energy optimization can be achieved on three levels. The system-level, the architecture-level, and the technological-level [2]. For instance, system-level optimization can be done by turning of the machine in inactive mode. For architecture-level optimization we can use parallel hardware to minimize interconnections which in turn reduce supply voltage without affecting the system-throughput. For technology-level optimization we can make changes to gate-level design. Figure below gives an overview of various energy optimization that can be considered for system design. For a design structure a designer has to consider various choices at different level. Parameters such as supply voltage and frequency is impacted by the selection of a particular algorithm. The most impact on the design structure can be achieved by modificatins at the highest level. Thus, very efficient and beneficial design choices can be made by optimizing algorithm and architecture at highest-level. However, during system system it is a challenge to anticipate the result and design decisions implications as implementation of the design depends upon the approximates at technology level and not at the higher level.

### C. Energy to Power Relationship

A significant relation between energy and power is that, the overall energy will not be affected much by reducing the amount of power and increasing the overall time taken by the tasks concurrently. This can be well understood by energy-power relationship as Power = Energy / Time [3]. The product of energy and delay can be used to achieve the goal of minimum power consumption for a standard power level.Power consumption by micro-processors are broadly classified into static and dynamic power consumption. Static power consumption refers to the amount of power consumed by the processor when there is no switching activity taking place. It is mainly a leakage current with fixed operating voltage. Static energy-utilization is a result of parameteres like the short circuit-currents, the configuered bias and leakage-currents [2]. The short circuit and switching power together constitute the Dynamic power consumed by the processor. Dynamic energy utilization (Pd) is a result of work done by the circuit to switch. It can be described as P =Pd + Psc + Pb + Pl [2]. Mostly CMOs technology is used for component assembly. As described two main reason for energy utilization on a CMOS chip, static and dynamic power consumption. The amount of energy required at the static level can be determined at the circuit-level. Both 'p' and 'n' channel devices will conduct together on changes at the input of the CMOS gate. This will result in a power wastage of 10 to 15%. Static power utilization is generally a small irrational portion of the dynamic power dissipation and thus more focus should be given to dynamic power optimization techniques. Following equations describe the switching power and circuit delay [3].

$$P\ (switching) = C.Vdd^2.a.f \qquad (1)$$

Where, C = Capacitance, Vdd = Voltage supply, a = activity factor, f = clock frequency.

$$\text{Delay} = \text{k.}\ \frac{Vdd}{(Vdd\text{-}Vth)^2} \qquad (2)$$

Where k= constant which changes with capacitance.

Major consumption of energy in CMOS is from the dynamic level and it can be noted that the dynamic power can be reduced by adjusting any of the above factors.

Power wastage is a result of periodic charging and discharging which is achieved by low to high logic-transition in a digital-circuit, this phenomenon results in energy utilization. The effective capacitance is a result of the capacitance which is charged or discharged, along with the possibility factor as shown by the equation Effective Capacitance = $\alpha$ C. Thus, a designer should minimize the parameters of this equation to reduce the energy wastage. However, research has shown that optimizing power usage is a subtle process of balancing variable in various tradeoff.

1) *Capacitance Reduction:* According to the equation (1) and (2) we can say that there are four alternatives to reduce power wastage viz. reduction in the capacitive-load (C), reduction in the supply-voltage (V), reduction the switching-frequency (f), reduction in the activity factor (a). CMOS circuitry capaciatnce is directly proportional on the energy utilization. Thus, to reduce energy utilization we need to reduce capaciatnce. The best place to achieve this by modifications at the technology level, However better results can be achieved by overall architecture design modification that utilize locality-of-reference [2]. Connection with the external elements usually have higer capacitance than on chip connections. Thus, to reduce the energy utilization we should not use more external components and reduce the switching between them. For instance, memory-access requires more energy. Thus, to reduce power consumption we should have less external memory access and use alternatives such as cache and registers. Routing capacitance is another reason for restrictions on clock frequency. Circuits with low routing capacitance can usually run faster, thus they utilize less power. Clock-frequency optimization can reduce the energy utilization. Another method to optimize capacitance is to reduce chip-area. Whereas just reducing chip area can sometimes lead to more energy utilization. For instance, an energy optimized design that had large on chip area can decrease the overall energy utilization by using parallel computing architecture [7].

2) *Voltage Reduction and Frequency Optimization:* One of the important options of minimizing energy consumption is reduction in the ammount of the supply voltage. Energy utilization reduces by a factor of four when we optimize the voltage usage. For instance, reducing voltage by 44% (5.1 to 3.2 volts) can reduce power utilization by about 55%. This the reason behind usage of low voltage systems by most of the processor vendors. However, reduction in voltage supply will causes the performance to degrade. Moreover, when supply voltage is optimized system clock must be reduced to ensure precise optimization.

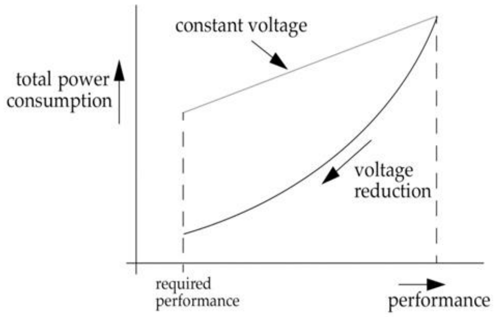


Fig 4. Voltage Scaling effect on power consumption [2]

Thus, any voltage reduction techniques should also cater for a drop-in performance. Extra hardware support is required to optimized throughput with reduction in supply voltage. Voltage and frequency optimization techniques have undesirable effect on energy consumption and time required for running a program. Reduction in clock-frequency alone will result in more energy utilization as execution time increases. Wait time increases when the voltage is reduced. A generic method to optimize power utilization is to performance improvements at the component level with additional hardware and reducing the voltage so that the required results are achieved (fig 3). Thus, the main idea behind low-power strategies is reduction of speed and critical-path so as to decrease the voltage requirement. Studies have shown that to reduce voltage usage, clock must be decreased.

*3) Activity Reduction Techniques:* The activity load from the equation-3 can be reduced by minimizing unwanted tie consuming activities. Some methods to describe them are discussed briefly as follows.

*a) Clock Control:* CMOS power utilization is directly dependent on the clock frequency. Switching off the clock is a common stratergy to minimize the power utilization. Clock Control can be obtained at hardware or at the application level. Some systems have sleep or idle mode [2] for clock control in which they switch off the clock to some components to reduce power consumption. When off the device does not function. Also, this design strategy of dynamic switching off the clock is a good candidate in designing synchronous machines.

*b) Transitions Control:* Energy utilization increseases with increase in the frequency of state change and the capacitance of the signal. This is applicable to every signal line of the system, for clock ticks, data path and address path. Thus, power utilization can be decreased by thorough check on the transition changes.

*c) Asynchronous Design:* Asynchronous design strategy is a novel technique to reduce power consumption. CMOS can achieve a great power reduction as gates need energy only when they switch to another state. Transition of the state should indicate that the gate is performing some useful work, whereas this is not always true in a synchronous circuit. Gates may change state even when they are not being doing some useful work on the input signal. The most important clock is the clock-driver that has to propagate clock ticks to all components of the circuit and thus it keeps on changing its state even when a small section of the circuit does work. So, a synchronous circuit dissipates power to a particular section of chip which is idle, for instance to a floating operation unit when integer computation is being done. Asynchronous system circuits are active only when performing some useful work and mostly operate on data. Section of asynchronous circuits which perform lesser data processing are provided lower power and subsequently the power consumption can be reduced.

*d) Reversible Logic:* In this technique, we try to persist data as long as possible so as to decrease energy utilization. Now a day's computers erase some of the data after they perform some operation. This deleted operation cannot be reverted. This data can be persisted and used in future when they are required thus reducing the computation requirement and also the power consumption. To achieve this, we can incorporate traditional methods such as have large storage devices like cache. These are called reversible systems and are difficult to implement.

## D. *Dynamic Voltage and Frequency Scaling Technique (DVFS)*

Dynamic scaling of the overall supply voltage to the processor and other components has a major effect in reduction of the overall energy dissipation. DVFS is considered to be a very effective technique to reduce the overall power consumption while maintaining performance statistics. This technique focuses on using the task and processor scheduling technique to slow down the processor while increasing the time taken by the processor to complete the assigned task. This helps the processor to operate with low power. This scheduling can be achieved by assigning priorities to the task either statically or dynamically. DVFS can be further achieved and classified as stated below [3]. Timing constraints technique: Scheduling based on this technique can be done on real time basis or non- real-time basis. Idea is to modify and change process and thread scheduling components of the operating system to achieve the same performance with considerable reduction in the amount of energy consumption. The results discussed as in [3] have an overall effect of 20%-40% reduction in energy consumption. Scaling Granularity Technique: This technique focuses on pre-emptive and non-pre-emptive techniques. In pre-emptive scheduling, equal time slice is distributed for each task by the scheduler where-as the non-preemptive scheduling technique consists of a group of periodic tasks which do not have any pre-emption and idle time constraints. Offline and Online Techniques: The offline techniques target the worst cases of a scheduling approach only thereby lowering power usage. Online techniques have a great potential to reduce power usage better than other techniques however at a cost of high processing overhead. Additional benefit of DVFS includes protection from power attacks.

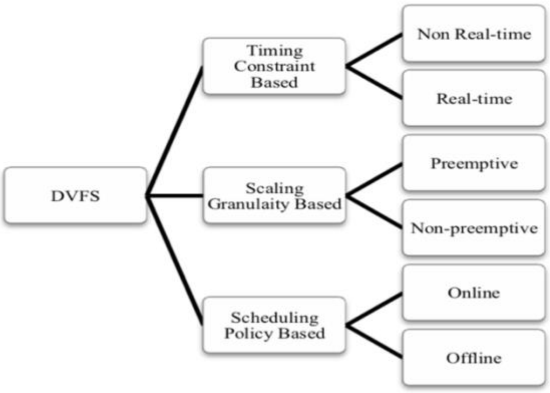


Fig 5. Classification Diagram for DVFS [3]

## E. *Clock and Power Gating Technique*

Reduction in clock frequency helps in reducing the dynamic power consumption by the processor. This directly refers to reduction in the amount of gate toggling which can be achieved in two ways viz. reducing total number of gates, reducing gate toggle frequency. The overall contribution of clock toggling in power consumption is about of 20-40% at the component level. Clock-Gating helps in power reduction in active mode of operation by 15% to 64% [3] as this technique directly reduces the capacitance by switching activity but it reduces the computational throughput and speed. Three categories of clock-gating include a) System-Level gating that focuses on disabling the systems by directly pausing the clock for a block. b) Sequential and combinational clock-gating has blocks continually providing output while optionally pausing the clock.

Power Gating is also considered as one of the most efficient techniques which directly targets in reduction of sub-threshold current leakage along with gate leakage. This is achieved by cutting off the supply voltage. In-order to provide sleep mode features in operating systems, this technique is heavily used. Below figure is a power gating technique using the Header-Footer switches.

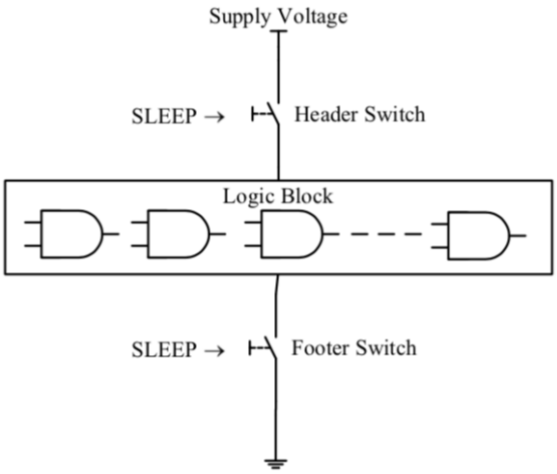


Fig 6. Power Gating [3]

### F. *Improvements at processor component levels*

This section of the paper describes power optimization and improvement techniques on various processor component like cache, pipelines, buses.

1) *Improvements at Cache levels*: In order to cope up with the pipeline clock frequency, the cache designs consist of tag and data arrays which are static RAMs, as a result the power dissipation due to caches is very high. Additionally, caches are populated with maximum number of transistors on the die resulting into high power dissipation of about 25% [3]. In case of bipolar multi-chips primary caches contribute to about 50% of overall power dissipation. Thus, caches are one of the most attractive components to achieve low-power consumptions, this can be done in many ways viz. improvements in memory cell designs, controlling voltage supply and overall structure redesign can optimize the power consumption by cache. Other novel techniques are described below.

*a) Addition of block-buffering:* Most of the low powered cache designs include a small additional cache or a buffer such that the data can be directly fetched form this buffer. Since the buffer is small it will significantly consume less power. This small system redesigning produces a significant amount of energy savings along with good performance as long as high hit rate can be ensured. Filter caches target the energy consumption by instruction issue and decode design. Experimental Analysis on several embedded systems shows an improvement of about 58 % power reduction while compromising about 21% performance.

*b) Cache-lines turned off:* This technique takes advantage of the memory cache hierarchy. Usually the lower level cache lines are busy and are called as Hot caches, whereas larger cache lines have a cold nature. This technique focuses upon putting the cold caches to sleep or low power mode until required while preserving the state in static rams. This can be achieved by following cache turn-off policies. [3][8] The inter-access time per cache access count technique turns of those cache lines which will not be reused, this reduces the leakage of L2 caches. The processes also involve marking cache lines invalid and removing invalid data, these also help in reducing the power consumption.

*c) Vertical-partitioning / Sub-banking technique:* Leakage power increases with decrease in the feature sizes, this has a direct impact over the structure of the cache since they live in a larger area. There is a significant improvement in the power consumption on subdividing the memory into smaller caches also knowns as sub-cache [8]. This technique also improves the cache hit ratio buy improving the locality of reference behavior [3]. Another alternative to internally dividing a single cache into smaller banks is to produce a multi-bank cache memory design. Horizontal partitioning eliminates unrequired cache accesses thereby contributing to low power.

*d) Improved Tagging scheme*: Cache tag comparisons are a good source of power consumptions as the entire tag array is compared in parallel, this becomes even worse in case of caches with high-associativity. A tag comparison is said to be unnecessary when the result of a match or mismatch can be easily determined at runtime without actual comparison Tag comparison-elimination is a novel technique that targets this behavior to reduce unrequired tag comparisons with a help of low-cost extra hardware. This has a significant effect on power reduction. As in case of set associative caches multiple sets and arrays of tags are compared in parallel, the use-way prediction [3] technique helps to achieve low energy consumption with high throughput. Experimental results show energy improvement by 60 to 70%. Other techniques involve modifications such as to compare only those instructions perform a fetch from a new buffer, along with modifications for tagging branches which resulted into a fetch from a new cache, the last modification involves use of S-cache along with on memory I-cache.

*2)* Improving instruction pipelines: Instruction pipelining employs instruction speculation to feed in the next instruction into the pipeline stages without the certainty of the flow of instruction. However, this speculative fetching turns out to consume a lot of energy in case of a branch instruction or when the instruction flow changes and the pipeline needs to be flushed. Following are the techniques which target improvements in the speculation as well as pipeline designing to reduce the overall power consumption.Usually pipeline designs employ the use of a large number of memory registers in-order to implement the bypass logic and enhance the throughput. However, these registers consume a lot of power, thus a significant decrease in their number will help in reduction of the overall power. Another technique known as pipeline-staging and unification(PSU) helps in reducing the power consumption by dynamically scaling the clock rate. The main object is to make use of by-passing registers to scale and unify different pipeline stage. Using PSU energy is typically saved by 1) reduction of the total number of pipeline stages by scaling the clock rate,2) Reduction of the driver’s overall load capacitance. Bipartition is a novel technique which involves dual-encoding in order to reduce the power consumed by the pipeline architecture [3]. As most of the energy consumption is due to the by-passing registers, the pipeline circuit is partitioned in a way that each of the smaller sub-circuits yield lesser number of varying outputs.

Moreover in-order to reduce the amount of register switching, encoding is employed on the logical blocks of both the smaller sub-partitioned circuitsAnother technique called "Pipeline-Gating" makes use of a confidence estimator to reduce the number of inaccurate speculation. This technique relies on the confidence estimate to decide if the branch prediction is accurate or not. The reduction in the number of inaccurate speculation saves a lot of energy and power. "Pipeline-balancing" [3] is a technique which focuses on dynamically manipulating the instructions issue rate using the differences in the instruction/cycle for every task. "Shallow pipelines" also yield a good result in reducing the amount of energy consumed by reducing the number of speculative fetching and instruction decoding

*3)* Improvements at Buses levels: Buses are a heavily used interconnect and signaling mechanisms. These can turn out to be a problem if not designed to be energy efficient. Usually a bus architecture consumes about 15-30% of total chip power [3] and this is ought to increase with technology scaling. Some novel techniques to reduce the power consumption by buses are discussed below.

*a)* Reduction in the bus swing: By adjusting the voltage swing on the bus wires the power usage by the interconnect system can be significantly reduced [3]. Bi-directional level converter circuit design of the driver is used as an interface to reduce and amplify the supply voltage at the transmitter and receiver respectively. The driver is responsible to convert a full-swing voltage signal into low-level reduced signal using the interface circuit and level up the signal at the receiver's end.

*b)* Bus Switching reduction by encoding: CMOS-Circuits usually make use of logic bit-values which turnout to consume a significant amount of power. Interconnected-capacitance in case of bus wires is quite high, subsequently capacitance on the wires can be reduced by adjusting the switching-activities. Bus-Encoding schemes focus in reducing the number of transitions in the interconnected circuit. Several-encoding schemes are employed based on the type of the bus, the "bus-invert" makes use of the difference in the hamming-distance between the new and the old data values, this helps in reducing the power consumption by 50%. Another "Gray-Code technique" makes use of the spatial-locality principle for addresses to reduce the switching-activity. Both of these techniques achieve reduction in the power usage by minimizing the Hamming-distance between data-addresses mediation operations. Additionally, there are significant efforts in nus-structure re-designing where the goal is to reduce the amount of switching-capacitance [3], one approach is bus-sub dividing into smaller bus components such that the overall capacitance gets distributed amongst smaller components thereby helping in reducing the power consumed by the interconnection architecture.

### *G. Modifications in the Issue-Logic Design*

Instruction issue-Logic is used to exploit ILP and it contributes to be one of the major source of power consumption. Issue logics make use of broadcast mechanisms to send wakeup signals during an associative search operation. This involves a large amount of tag bit comparisons which consume a lot of power. About 18% of power consumption by associative search operations was recorded for Alpha 21264 System [1]. Several novel methodologies are proposed below to make the issue logic design power efficient, the main focus is to make use of indexed table to implement the wake-up mechanism and not make use of associative search thereby significantly draining down the power consumption. First Scheme is a Dependence-Tracking Scheme, which suggest two techniques [1].

*1) N-use Issue technique [1]:* This technique exploits the fact that registers are usually read for few times, thus it makes use of an N-use table which contains the starting n-instructions for every register which is read in a proper-sequential order. The techniques make use of two hardware based components. The first one is the n-issue table which is used to keep track of every instruction which is not read and the second one an instruction-ready queue which contains all ready instructions which have all the required operands available, this queue is responsible for in-order issuing of instructions. Once an instruction is done execution the register identifier is used to consult the table, if there exists an instruction, the pointer to that field is taken under consideration otherwise the ready queue is the residence for this instruction. The scheme gives an enhanced performance when augmented with the use of an I-buffer [1]. By avoiding the associative tag comparisons and using the n-issue table a lot of power is saved.

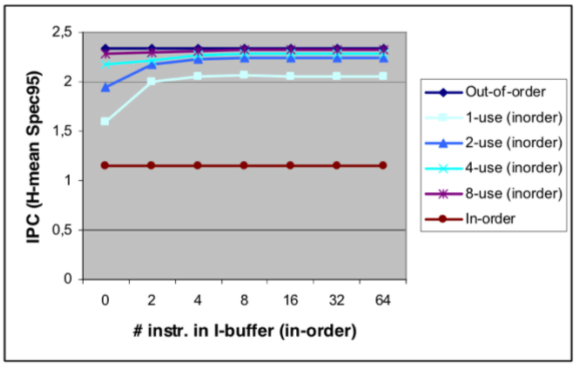


Fig.7. Performance Comparison of N-Use with I-Buffer wait queue with out of order scheme [1]

*2) The distance-Issue scheme mechanism [1]:* This scheme focuses on instruction scheduling based on the latencies after instruction decode. This scheme also employs extra hardware components to store the instructions until all the required values are available. It typically consists of a table to check register availability, waiting queue and instruction issue queue. The main concept is that instructions are separated by distance and are executed in the proper order. The distance is used to ensure that instruction dependencies are followed. The wait queue is used to the instructions in case it is depending on a load operation, on a mem-write back operation the registers are marked available with the desired data and this register is then broadcasted to the wait queue to wake up other dependent instructions. Other approach is a Deterministic-Latency approach, where all the memory operations are considered to have a constant latency, due to which all the instructions in the read queue are ordered according to the predicted latencies. However, this does not avoid instruction stalling when the dependent instruction is starving for the write-back of previous dependent instruction. This technique maintains the cycle information of when the values will be produces for every physical register, it also has a delayed issue structure to maintain instructions which were issued before. In this scheme, the instructions are issued like any other technique but is assumed to have a latency time equivalent to the hit time of level-1 cache.

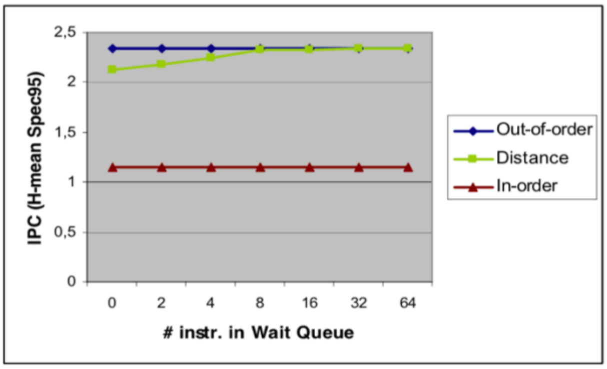


Fig.8. Performance Comparison of Distance issue scheme with wait queue with out of order scheme [1]

The figure below shows the performance comparison chart for all the techniques discussed above, It can be noted that the performance degradation is insignificant as compared to out of order execution, while the energy savings is quite significant

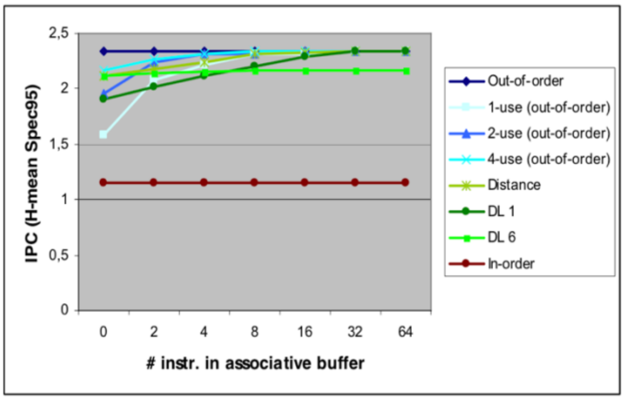


Fig.9. Performance Comparison of all techniques [1]

The schemes proposed make use of less number of associative structures and prove to be more energy efficient than out of order scheme the graph below shows the percentage energy savings by all the proposed techniques.

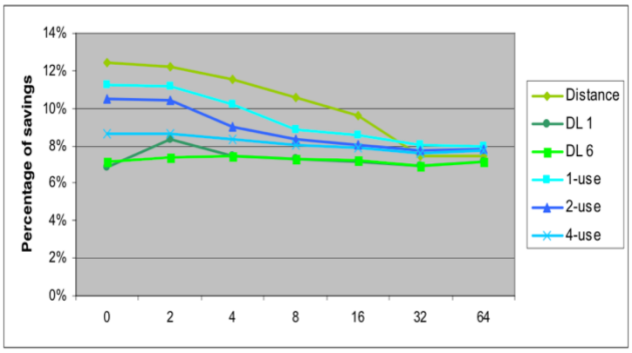


Fig.10. Energy savings chart [1]

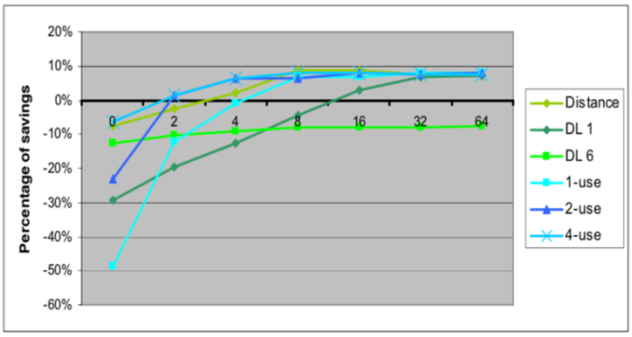


Fig.11. Energy-Delay product chart [1]

### *H. System Architecture Level Optimization*

This section mainly throws light on the low power optimization techniques at system level. Two main factor which needs to be considered for system design are reducing unwanted operations and making use of locality of reference. Studies have shown that 10 to 40% pf the total energy dissipation is due to interconnected between components buses, multiplexers and drivers. The element that uses large amount of power is the interconnect. Energy utilization can increase by large factor with multiple chips. Two main parameters for reducing the energy consumption due interconnecting components if locality and uniformity. Locality is the amount of separation between various operation or components [2]. Dividing the system/algorithm into cluster with maximum communication so that most of the transfer of data takes place between the cluster and not across clusters. This will result is more utilization of the local-bus than the high-power dissipating global-buses. This reduces hardware sharing as communicating interconnection are reduced by a large factor. The methods create a closely packed layout with a smaller number of multiplexer/buffers. Uniformity in an algorithm indicates the repetition of computational patterns. Repetition allows design to be less complicated structure and hence simple architecture and less control hardware. Following are the mechanism that use locality of reference as a tool to achieve reduction in energy utilization [12].

*1) Application-specific package modules*: Locality decreases the amount to of power consumed in. communication and enables use of small transistors, which reduced the overhead of capacitance. Localization is done using cache and pipeline. Additionally communication overhead can be reduced by combining the processor and memory [10,11]. For system level locality we can separate the functionality of the design into different modules with similar traits [12]. If the system is divided into separate dedicated co-processors the communication traffic can be decreased. An example scenario can be to stream video on the screen, the content can be sent directed directly to the memory designated for screen instead of sending it via the main processor [4]. Sometimes processor have to complete computations that are not best suited for them. Even though they can do these jobs it will take them longer the usual and also utilize more power. The best solution for this is Application specific integrated circuits (ASIC). They have dedicated computing units placed around a normal processor. Designed can use these processors for some section of the algorithm for which it is best suited and used the normal process for the rest of the work. It is a good example of a processor which may utilize more power than a normal but will accomplish same task in far less time than the normal process resulting in an overall power saving. By thorough factoring of a system we can not only reduce the power utilization but also increase the performance [9].

*2) Hierarchical memory systems:* Multi-level memory system can be used in computing machines to reduce power utilization. Such a system is called as hierarchical memory system. The main purpose of such a system is to store regularly used piece of code in a small memory closer to the processor. Energy utilization is minimized as majority of the time this small memory is used as the read from the external memory is avoided. Thus, having an on-chip cache will significantly decrease the power consumption and increase the performance to the system [2]. Communication traffic is reduced as the information is stored locally. This technique can also be applied to file system optimization. Maximum reduction in energy utilization can be achieved by using register for data storage. As compared to read operation write requires more amount of energy as a write operation always causes an off-chip memory access.

*3) Value compression technique for Ultra-Low Power Designs:* Value compression is a novel and effective technique to reduce consumption for Ultra-Low power computers. It exploits the fact that the energy utilized by the data paths depend upon the values in the pipeline stages. This technique makes use of lesser number of average bits which are required to represent the values altered in the pipeline [2].

*4) The Pocket Companion's system architecture:* This architecture aims at improving the system performance as well as optimization of the overall energy utlization by the systems. The complexity of incorporating all above requirements in on design arises from the goal of power optimization with performance improvement [2].

Pocket Companion's architecture focuses on optimizing the energy utilization and not only on system performance. This technology not only reduces energy consumption but also focuses on providing enchanced services including multimedia devices, compression, network access, deompression and security functions

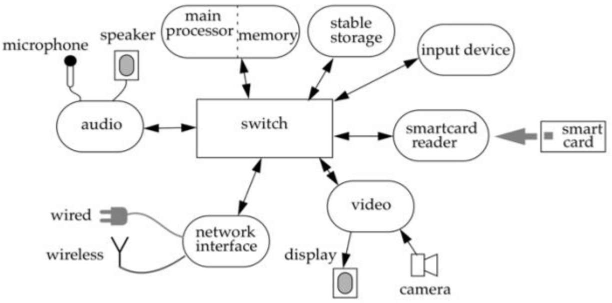


Fig.12. System Design

These designs have non-traditional design which use system-decomposition at various levels of architecture to optimize energy consumption. This architecture has various common purpose processor along with a bunch of different programmable modules, each has a optimized implementation of a particular function. For instance, application specific processors can complete some operation faster with less power utilization than general purpose processor [2]. It has a reconfigurable internal communication-network that used locality principle to store data for a long time. This design also reduced unwanted operations by using dedicated functional unit that can be individually turned off and are triggered by data. Every package-module has a separate clock voltage control, power off modes.

This architecture differentiates a switch by surrounded components. All functional units in the system communicate via a communication network. The switch acts a connection between different units and establishes a reliable communication channel. Like a switch network, use of different switches will facilitate parallel information sharing and thus will increase the performance. Modularity in functional unit will reduce the communication and the involvement of processor. Thus, these components have less power dissipation and have their own energy management [6].

## III. Conclusion

Attention has to be given to optimizing power utilization as there will be more progress on portable hand-held devices.

Architect can reduce the power consumption by using abstraction at hardware and software level. This paper has some valuable techniques as well as architecture for achieving the goal of optimized energy utilization. For technology level energy utilization can be reduced by decreasing the supply-voltage, capacitance and by switching off the power supply when not in use. More benefits can be achieved by reducing the unwanted activities at both architectural and system-level. DVFS technique is discussed where task-based scheduling can be used to reduce the clock-speed and supply voltage. We have also stated two important optimization techniques "dependency-tracking" and "Prescheduling". In dependency tracking we try to maintain the reliance of instruction on one another in table format on a physical register. Use of distance-scheme and deterministic-latency techniques can achieve same performance as the out-of-order processor. For system-level power optimization, we can make use of power-management techniques, at the same time we can also differentiate system component based on their functionality and programming techniques for optimizing power utilization. It is evident for the stated discussion that some energy optimization techniques not only reduce power utilization but also increase performance. In general, sophisticated code take less time and space and therefore will reduce energy consumption. Using components such has cache, coprocessors can optimize the performance of the system with respect to various parameters. The proven advantage of using cache is to reduce off chip access which decrease delay as well as bus power. Improvements at the pipeline level is also helpful in optimizing the component level power utilization. The approach of using synchronous system can also reap high performance as the speed of the execution is no longer based on the clock, but at the cost of performance. Energy-cost efficient system can be complex and may require extra hardware for robust implementation.

## Acknowledgment

This work in this section is supported by studies in the UK Engineering and Physical Sciences Research Council (EPSRC) under Grants EP/C005686/1, EP/C014790/1 and EP/C54630X/1. We want to express gratitude towards Dr. Ehat Ercanli for his valuable counselling and guidance, and for encouraging us to work on this topic. We also want to acknowledge our colleagues for their critique and constructive feedback on our work anonymously.

## References

[1] Ramon Canal Corretger, “Power- and Performance- Aware Architectures“. http://personals.ac.upc.edu/rcanal/pdf/PowerAndPerformanceAwareArchitectures.pdf

[2] Paul J.M. Havinga, Gerard J.M. Smit on, “Design techniques for low power-systems”. http://wwwhome.cs.utwente.nl/~havinga/papers/energy.design.techniques.jsa.pdf

[3] Muhammad Yasir Qadri, Hemal S Gujarathi and Klaus D. McDonald-Maier on, “Low Power Processor Architectures and Contemporary Techniques for Power Optimization – A Review”. https://core.ac.uk/download/pdf/16470451.pdf

[4] Enric Herrero1, Jos´e Gonz´alez2, and Ramon Canal on, “Power-Efficient Spilling Techniques for Chip Multiprocessors”. http://people.ac.upc.edu/rcanal/pdf/Herrero-EuroPar-2010.pdf

[5] Design techniques for energy efficient and low-power systems. http://wwwhome.cs.utwente.nl/~havinga/thesis/ch2.pdf

[6] Thomas D. Burd and Robert W. Brodersen on, “Energy Efficient CMOS Microprocessor Design “.

https://pdfs.semanticscholar.org/9339/487170b8eac7664f53f969c32d5a2268f99d.pdf

[7] BILL MOYER on, “Low-Power Design for Embedded Processors”.

http://www.ruf.rice.edu/~mobile/elec518/readings/ProcessorArchitecture/moyer00proceedings.pdf

[8] Matteo Monchiero and Ramon Canal and Antonio Gonzalez on, “Using Coherence Information and Decay Techniques to Optimize L2 Cache Leakage in CMPs”.

http://people.ac.upc.edu/rcanal/pdf/monchiero-icpp09.pdf

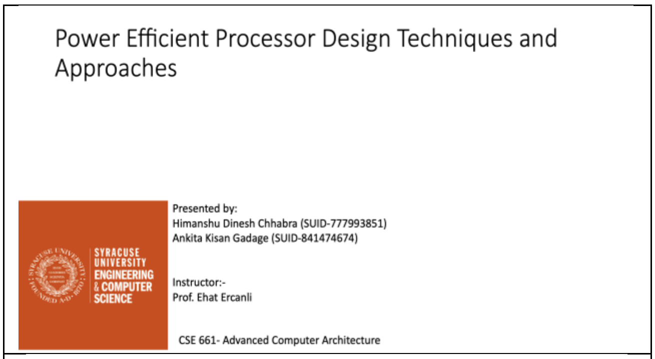
[9] Mangione-Smith, W., et al.: “A low power architecture for wireless multimedia sys- tems: lessons learned from building a power hog”, Proceedings of the international sympo- sium on low power electronics and design, Monterey, August, 1996

[10] “A Case for Intelligent DRAM: IRAM”, Hot Chips 8 A Symposium on High-Performance Chips, information can be browsed on: http://iram.cs.berkeley.edu/publications.html

[11] McGaughy, B: “Low Power Design Techniques and IRAM”, March 20, 1996, information can be browsed on http://rely.eecs.berkeley.edu:8080/researchers/brucemcg/iram_hw2.html

[12] Abnous A, Rabaey J.: “Ultra-Low-Power Domain-Specific Multimedia Processors,” Pro- ceedings of the IEEE VLSI Signal Processing Workshop, San Francisco, October 1996.

## APPENDIX



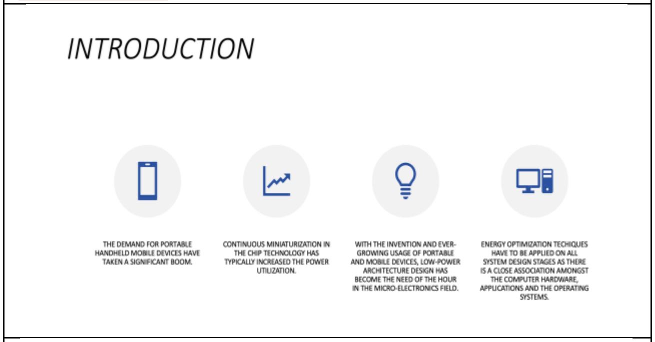
Power Efficient Processor Design Techniques and Approaches

SYRACUSE UNIVERSITY
ENGINEERING & COMPUTER SCIENCE

Presented by:
Himanshu Dinesh Chhabra (SUID-777993851)
Ankita Kisan Gadage (SUID-841474674)

Instructor:-
Prof. Ehat Ercanli

CSE 661- Advanced Computer Architecture



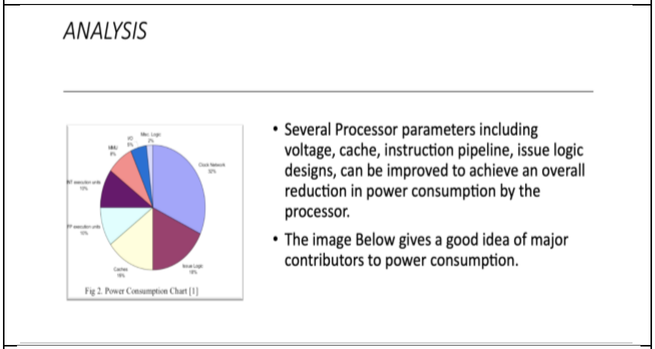
INTRODUCTION

THE DEMAND FOR PORTABLE HANDHELD MOBILE DEVICES HAVE TAKEN A SIGNIFICANT BOOM.

CONTINUOUS MINIATURIZATION IN THE CHIP TECHNOLOGY HAS TYPICALLY INCREASED THE POWER UTILIZATION.

WITH THE INVENTION AND EVER-GROWING USAGE OF PORTABLE AND MOBILE DEVICES, LOW-POWER ARCHITECTURE DESIGN HAS BECOME THE NEED OF THE HOUR IN THE MICRO-ELECTRONICS FIELD.

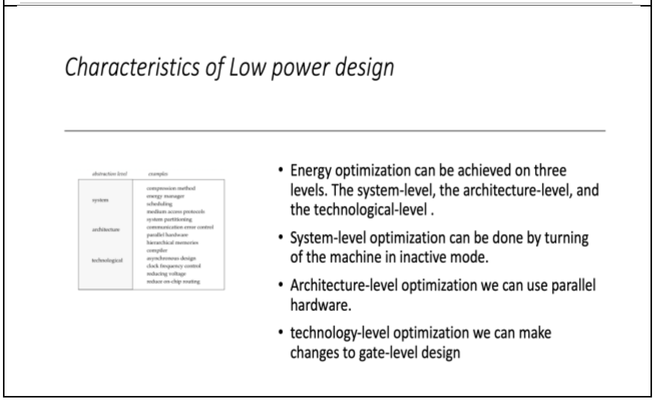
ENERGY OPTIMIZATION TECHIQUES HAVE TO BE APPLIED ON ALL SYSTEM DESIGN STAGES AS THERE IS A CLOSE ASSOCIATION AMONGST THE COMPUTER HARDWARE, APPLICATIONS AND THE OPERATING SYSTEMS.



ANALYSIS

Fig 2. Power Consumption Chart [1]

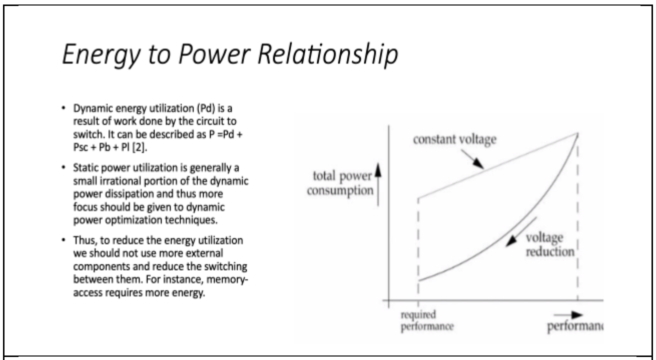
- Several Processor parameters including voltage, cache, instruction pipeline, issue logic designs, can be improved to achieve an overall reduction in power consumption by the processor.
- The image Below gives a good idea of major contributors to power consumption.



Characteristics of Low power design

| *abstraction level* | *examples* |
|---|---|
| system | compression method<br>energy manager<br>scheduling<br>medium access protocols |
| architecture | system partitioning<br>communication error control<br>parallel hardware<br>hierarchical memories<br>compiler |
| technological | asynchronous design<br>clock frequency control<br>reducing voltage<br>reduce on-chip routing |

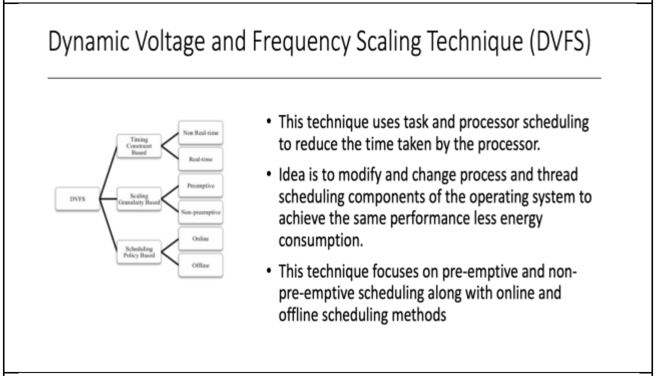
- Energy optimization can be achieved on three levels. The system-level, the architecture-level, and the technological-level .
- System-level optimization can be done by turning of the machine in inactive mode.
- Architecture-level optimization we can use parallel hardware.
- technology-level optimization we can make changes to gate-level design



## Energy to Power Relationship

- Dynamic energy utilization (Pd) is a result of work done by the circuit to switch. It can be described as P =Pd + Psc + Pb + Pl [2].
- Static power utilization is generally a small irrational portion of the dynamic power dissipation and thus more focus should be given to dynamic power optimization techniques.
- Thus, to reduce the energy utilization we should not use more external components and reduce the switching between them. For instance, memory-access requires more energy.

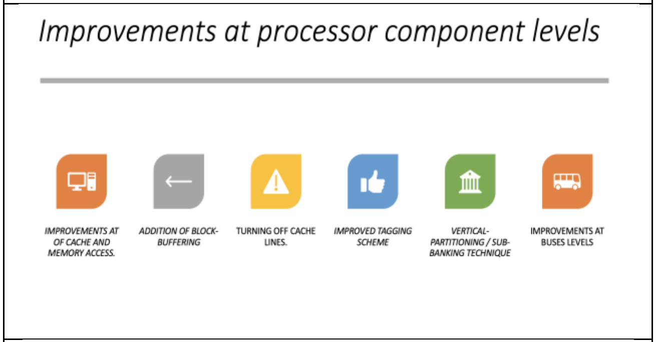
constant voltage
total power consumption
voltage reduction
required performance
performanc



## Dynamic Voltage and Frequency Scaling Technique (DVFS)

DVFS
Timing Constraint Based
Scaling Granulaity Based
Scheduling Policy Based
Non Real-time
Real-time
Preemptive
Non-preemptive
Online
Offline

- This technique uses task and processor scheduling to reduce the time taken by the processor.
- Idea is to modify and change process and thread scheduling components of the operating system to achieve the same performance less energy consumption.
- This technique focuses on pre-emptive and non-pre-emptive scheduling along with online and offline scheduling methods



## Improvements at processor component levels

*IMPROVEMENTS AT OF CACHE AND MEMORY ACCESS.*

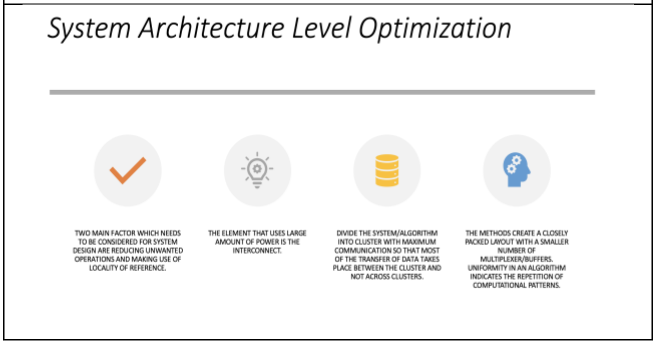
*ADDITION OF BLOCK-BUFFERING*

TURNING OFF CACHE LINES.

*IMPROVED TAGGING SCHEME*

*VERTICAL-PARTITIONING / SUB-BANKING TECHNIQUE*

IMPROVEMENTS AT BUSES LEVELS



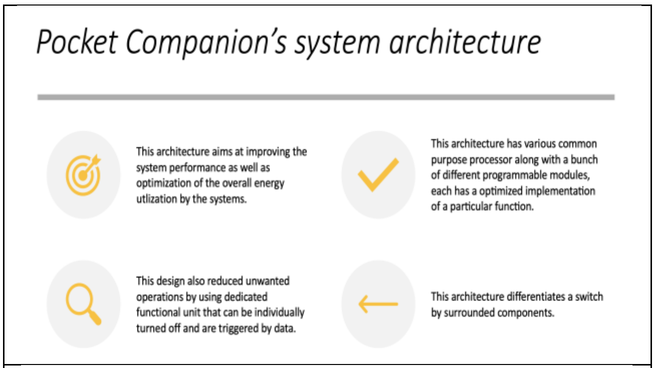
## System Architecture Level Optimization

TWO MAIN FACTOR WHICH NEEDS TO BE CONSIDERED FOR SYSTEM DESIGN ARE REDUCING UNWANTED OPERATIONS AND MAKING USE OF LOCALITY OF REFERENCE.

THE ELEMENT THAT USES LARGE AMOUNT OF POWER IS THE INTERCONNECT.

DIVIDE THE SYSTEM/ALGORITHM INTO CLUSTER WITH MAXIMUM COMMUNICATION SO THAT MOST OF THE TRANSFER OF DATA TAKES PLACE BETWEEN THE CLUSTER AND NOT ACROSS CLUSTERS.

THE METHODS CREATE A CLOSELY PACKED LAYOUT WITH A SMALLER NUMBER OF MULTIPLEXER/BUFFERS. UNIFORMITY IN AN ALGORITHM INDICATES THE REPETITION OF COMPUTATIONAL PATTERNS.



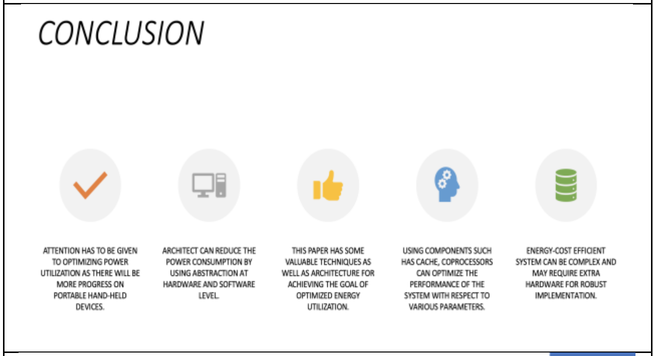
## Pocket Companion's system architecture

This architecture aims at improving the system performance as well as optimization of the overall energy utilization by the systems.

This architecture has various common purpose processor along with a bunch of different programmable modules, each has a optimized implementation of a particular function.

This design also reduced unwanted operations by using dedicated functional unit that can be individually turned off and are triggered by data.

This architecture differentiates a switch by surrounded components.



## CONCLUSION

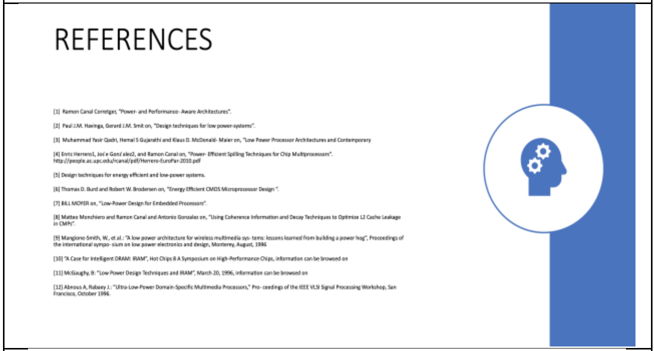
ATTENTION HAS TO BE GIVEN TO OPTIMIZING POWER UTILIZATION AS THERE WILL BE MORE PROGRESS ON PORTABLE HAND-HELD DEVICES.

ARCHITECT CAN REDUCE THE POWER CONSUMPTION BY USING ABSTRACTION AT HARDWARE AND SOFTWARE LEVEL.

THIS PAPER HAS SOME VALUABLE TECHNIQUES AS WELL AS ARCHITECTURE FOR ACHIEVING THE GOAL OF OPTIMIZED ENERGY UTILIZATION.

USING COMPONENTS SUCH HAS CACHE, COPROCESSORS CAN OPTIMIZE THE PERFORMANCE OF THE SYSTEM WITH RESPECT TO VARIOUS PARAMETERS.

ENERGY-COST EFFICIENT SYSTEM CAN BE COMPLEX AND MAY REQUIRE EXTRA HARDWARE FOR ROBUST IMPLEMENTATION.



## REFERENCES

[1] Ramon Canal Corretger, "Power- and Performance- Aware Architectures".

[2] Paul J.M. Havinga, Gerard J.M. Smit on, "Design techniques for low power-systems".

[3] Muhammad Yasir Qadri, Hemal S Gujarathi and Klaus D. McDonald- Maier on, "Low Power Processor Architectures and Contemporary

[4] Enric Herrero1, Jos'e Gonz'alez2, and Ramon Canal on, "Power- Efficient Spilling Techniques for Chip Multiprocessors". http://people.ac.upc.edu/rcanal/pdf/Herrero-EuroPar-2010.pdf

[5] Design techniques for energy efficient and low-power systems.

[6] Thomas D. Burd and Robert W. Brodersen on, "Energy Efficient CMOS Microprocessor Design ".

[7] BILL MOYER on, "Low-Power Design for Embedded Processors".

[8] Matteo Monchiero and Ramon Canal and Antonio Gonzalez on, "Using Coherence Information and Decay Techniques to Optimize L2 Cache Leakage in CMPs".

[9] Mangione-Smith, W., et al.: "A low power architecture for wireless multimedia sys- tems: lessons learned from building a power hog", Proceedings of the international sympo- sium on low power electronics and design, Monterey, August, 1996

[10] "A Case for Intelligent DRAM: IRAM", Hot Chips 8 A Symposium on High-Performance Chips, information can be browsed on

[11] McGaughy, B: "Low Power Design Techniques and IRAM", March 20, 1996, information can be browsed on

[12] Abnous A, Rabaey J.: "Ultra-Low-Power Domain-Specific Multimedia Processors," Pro- ceedings of the IEEE VLSI Signal Processing Workshop, San Francisco, October 1996.

Thank you