Low Power Techniques in Digital Systems

SOCRATES'04

Joan Oliver

ETSE-UAB

Power Analysis Estimation on VLSI Circuits

- Actual microprocessor trends in power consumption
- Sources of power consumption in CMOS circuits
  - Power consumption in CMOS circuits
  - Delay in MOS devices
  - Scaling principles for low power
  - Architecture driven voltage scaling

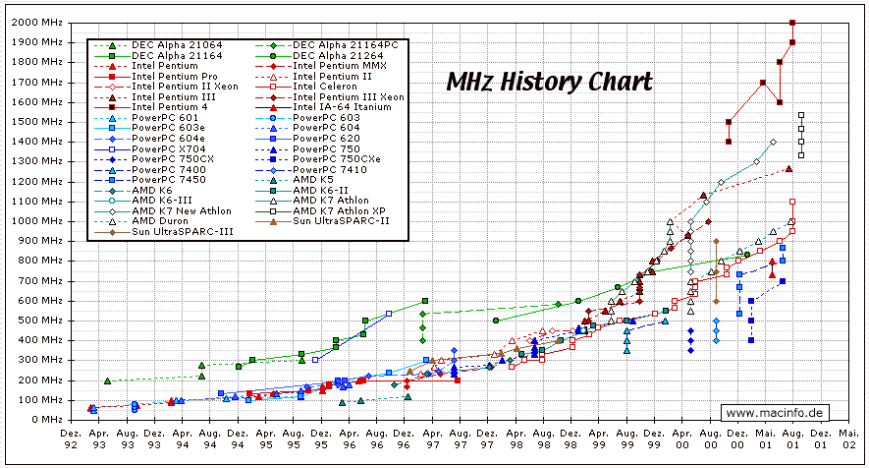
Power consumption microprocessor trends

- The continuing decrease in feature size and the corresponding increase in chip density and operating frequency have made power consumption a major concern in VLSI design. Modern microprocessors and microcontrolers are indeed hot. For exemple, following tables show the evolution in terms of power consumtion and clock speed of the most used microprocessors.
- In 1965 Gordon Moore stated that complexity of microprocessors will double every 12 months. This has been to be adjusted to every 24 months, but...
- The data published in January 2003 of the National Roadmap for Semiconductors predicts the continuation of this trend. For instance, current microprocessors have feature sizes (in gate lenghts) of 37nm and speed clock in microprocessor of 4.2GHz. This specs tend to be reduced to 7nm, and that microprocessor clock speed tends to 53GHz by the year 2018.

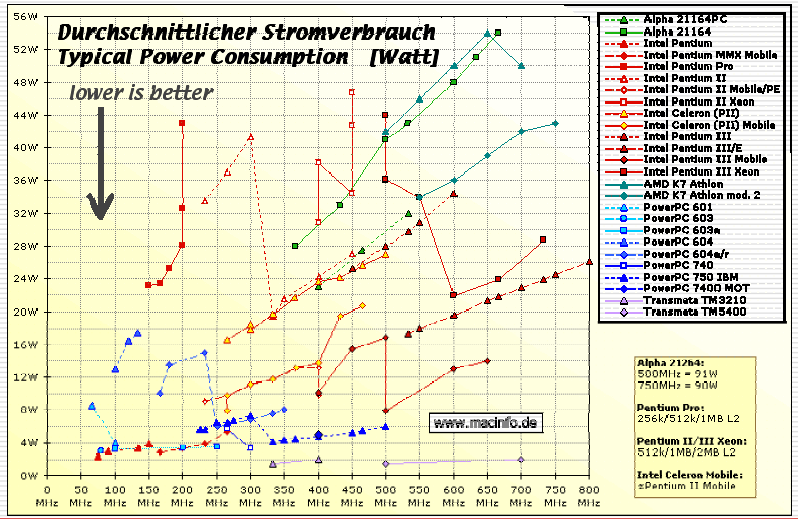
ITRS Technology Nodes and Chip Capabilities[2]

	2004	2007	2010	2018
DRAM Half-Pitch (nanometers)	90	65	45	18
DRAM Memory Size (mega or gigabits)	1G	2G	4G	32G
DRAM Cost/Bit (micro-cents)	2.7	0.96	0.34	0.021
Microprocessor Physical Gate Length (nanometers)	37	25	18	7
Microprocessor Speeds (GHz)	4.2	9.3	15	53

Power consumption microprocessor trends



Power consumption microprocessor trends



CPU	MHz	SPECint95 (Besser →)	SPECfp95 (Besser →)	Strom/Watt (← Besser)	Jahr
PowerPC 603e	300	7.7	6.1	(250MHz) 3.5W	1997
PowerPC 604	180	6.2	5.3	(120MHz) 16.5W	1996
PowerPC 604e	233	10.3	7.3	16.7W	1997
PowerPC 604r	375	15.9	10.1	(350MHz) 8.0W	1997
PowerPC 740	300	12.2	7.1	3.4W	1998
PowerPC 750	500	23.9	14.6	6.0W	1999
PowerPC X704	533	ca. 12	ca. 10	~80W	1997
PowerPC 7400	450	21.4	20.4	(400MHz) 5.0W	*1999*
Pentium	200	5.2	4.3	(120MHz) 10.xW	1995
Pentium MMX	233	7.1	5.2	17.xW	1997
Pentium Pro/256k	200	8.2	6.2	28.xW	1996
Pentium Pro/1MB	200	8.7	6.8	43.xW	1997
Celeron	500	17.9	12.9	27.0W	1999
Pentium II	450	17.2	12.9	27.1W	1998
Pentium II Xeon/2MB	450	19.7	15.0	46.7W	1998
Pentium III	600	24.0	15.9	34.5W	1999
Pentium III/E	800	38.4	28.9	26.2W	1999
Pentium III Xeon/512k	550	23.6	16.9	34.0W	1999
AMD K7 Athlon	750	32.8	24.3	50W@700MHz	1999
Alpha 21164PC	583	16.7	20.7	~45W	1997
Alpha 21164	667	20.8	32.4	54W	1998
Alpha 21264	667	31.8	49.0	(600MHz) 109W	1999
HP PA-RISC 8000	180	11.8	20.2	~40W	1996
HP PA-RISC 8200	240	16.4	25.3	???	1997
HP PA-RISC 8500	440	34.0	51.4	???	1998
Sun UltraSPARC II	450	19.6	27.1	(250MHz) ~25W	1998
Sun UltraSPARC IIi	360	15.2	19.9	???	1998
Sun UltraSPARC III	600	35++	60++	???	1998
SGI MIPS R10000	250	14.7	24.5	(180MHz) ~30W	1997

Power consumption microprocessor trends

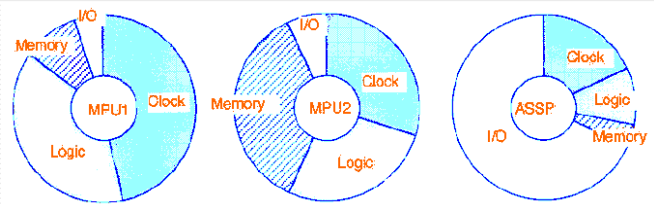
- Smaller festure sizes → integration of a larger number of components in a chip
  - reduction of the signal propagation delays
  - higher clock frequencies.
  - But, ... overall increase of power dissipation,
    - → overheating
    - degrades performances
    - reduces chip lifetime
    - portability problems.
  - → Identification of low power design as a technological critical need.
- For microprocessors there exist an analytical relationship between power consumption, area and clock frequency that potentially limits integration density:

$$P_{\mu P} = 0.063 \frac{W}{cm^2 MHz} A f_{clk}$$

- There is a strong necessity for minimising power consumption when designing complex microelectronic digital circuits and systems:
  For example, portable computation and wireless communication devices imposes very tight restrictions on the design to minimise power consumption, at the same time that real time digital processing (data, video, audio) require high computational resources to meet the requirements of the process.

Power consumption microprocessor trends

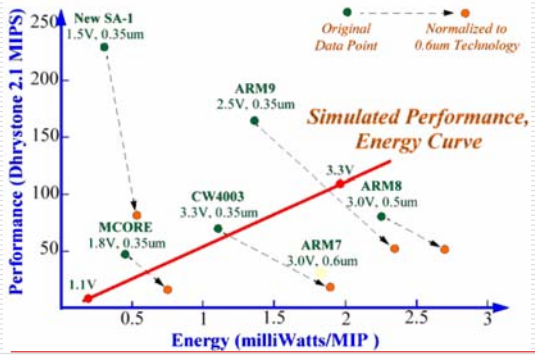
- Actual design tools supply designers with advanced tools, from the system specification to the mask layout, with stepwise refinement processes, that allows the specifications at each stage to be optimised using tools at different level of abstraction.
- But, the components that contribute to the overall power consumption differs from component to component. That is, power optimisation is an inherently application specific problem to be carefully analysed for each component.



Power distribution of three designs: low-end microprocessor for embedded use, high-end CPU and MPEG2 decoder

Power consumption microprocessor trends

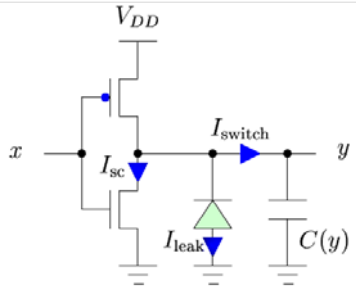
- Requirements of portability places severe restrictions on size, weight, and power.
- Power is important in that conventional nickel-cadmium battery technology provides about 20 W.hrs of energy for each pound of weight. Actually, notebook and laptop computers,... are demanding the same capabilities as found in desktop machines.
- Portability is no more associated with low-throughput. Low power is a crucial requirement in actual required environment.



Power consumption in embedded cores

Power consumption sources in CMOS circuits

$$P_{avg} = P_s + P_{sc} + P_d = I_{leak} \cdot V_{DD} + I_{sc} \cdot V_{DD} + \alpha \cdot C_L \cdot f_{clk} \cdot V_{DD}^2$$



- Leakage power is due to substrate injection in the p-n junctions and subthreshold effects.
  Contribution usually less than 1% of the P_{avg}.
  Becomes important when working in ultra-low power systems with very small VDD. It should be also considered when the chip is going to be standby mode for most of the time.
- Short-circuit power. Caused by currents that temporarily occur when parts n and p of the gate are open during the switching of the gate.
  In a switching CMOS inverter with PMOS and NMOS devices with threshold voltages V_{tp} and V_{tn} is present when $V_{DD} > |V_{tp}| + |V_{tn}|$, and its value increases proportionally with the rise time of the input.
  Typically, responsible of the 10% to 20% of the overall power consumption.

Power consumption sources in CMOS circuits

- Capacitive or switching power. The most significant component power contribution (accounts for about the 80% of the total power consumption).
  It is the power dissipated by the current that loads the capacitance of the output node of the gate.
  Total switching power of a circuit can be expressed as

$$P_{sw} = \frac{1}{2} f_{clk} V_{DD}^2 \sum_{signal\ y} \alpha(y) C(y) = C_{eff} f_{clk} V_{DD}^2$$

- The most important contribution to power consumption. Can be reduced
  - Reducing the power supply voltage. But $f_{clk} \propto 0.7 - \frac{V_t}{V_{DD}}$
  - Minimising the fanin logic feeding signals y.
  - Reducing the physical capacitance C(y) to be switched by signal y and the wiring capacitance. It is technology dependent.
    But wiring length does not scale down proportionally to the feature sizes of the technology.

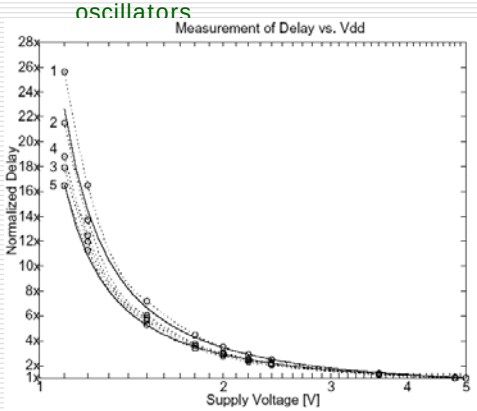
Power consumption sources in CMOS circuits

- Delay in MOS devices
  - One of the effective way of reducing power in MOS devices is lowering the supply voltage. But delay in MOS devices increases when decreasing the supply voltage.
  - For long channel devices $t_d = \frac{Q}{I_D} = \frac{C_L V_{DD}}{I_D} = k' \frac{C_L V_{DD}}{(V_{DD} - V_t)^2}$
  - For short channel devices, $I_D = (V_{DD} - V_t)^{\alpha}$

    with α = 1.3 for sub-half micrometer MOSFET's for a wide range of technologies from L_{eff} = 0.4μm down to L_{eff} = 0.1μm
  - Then $t_d = \tilde{k} \frac{C_L V_{DD}}{(V_{DD} - V_t)^{\alpha}}$

Power consumption sources in CMOS circuits

- Normalised delay of the some circuit primitives (transmission gate full adder, standard cell mux, full adder based on standard cell, standard cell inverter) configured as ring oscillators

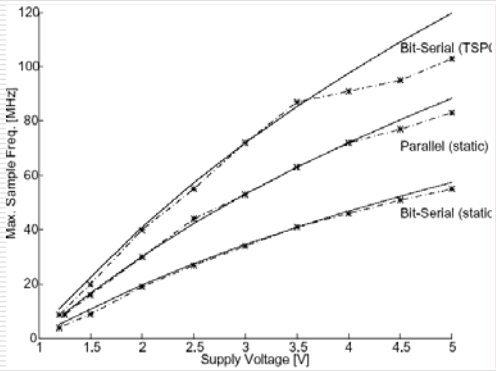


Measured delay (normalised to 1.0 at V_{DD}=5V) vs. supply voltage (0.8µm, V_{tn}=0.8)

Power consumption sources in CMOS circuits

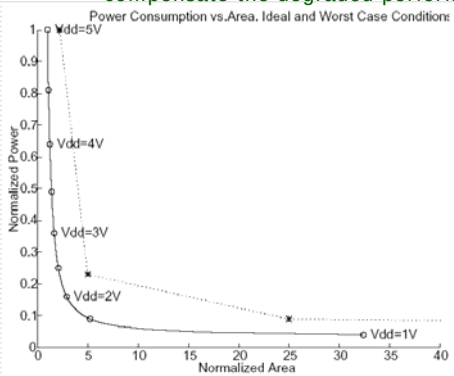
- Applications of the model to complex architectures.

  Dotted line curves shows test chip performance measurements (f_s = k"/t_d) made to three different wave digital filters architectures. Solid line curves represent performance expressed as the inverse of the simple delay model



Power consumption sources in CMOS circuits

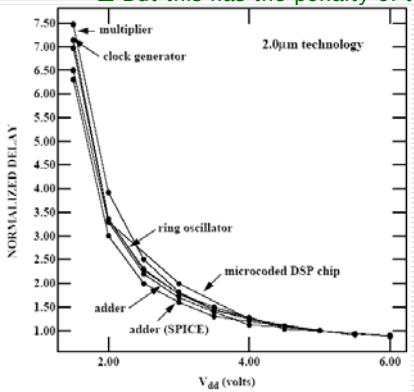
- The analysis of power consumption for constant throughput systems implies the reduction of the power supply as much as possible.
- Supply voltage reduction minimises power. Chandrakasan, Brodersen et alt. propose the use of parallelism and pipeline as a means to compensate the degraded performance due to power supply lowering



- Since the number of elements N is proportional to the delay, the ordinate axis represents the number N of parallel processing elements needed at a given supply voltage to give the same performance as one element at 5V.
- Since the silicon area is proportional to the number N, the cost of the power consumption reduction is a factor N increase in silicon area.
- This trade-off power-area is represented in the figure. Figure also plots the worst case area (*) for some supply voltage values

Power consumption sources in CMOS circuits

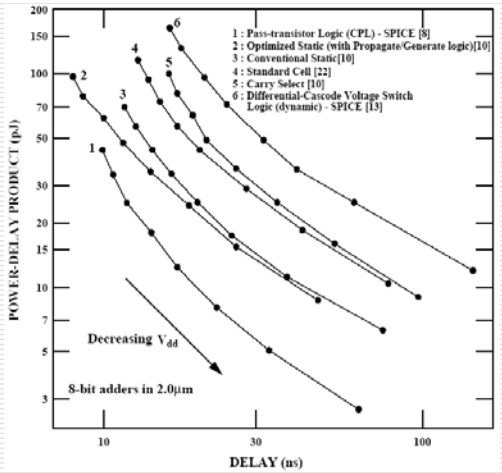
- Scaling principles for low power
  - The reduction of V_{DD} should yield great benefits (switching power consumption = $C_{eff} \cdot f_{clk} \cdot V_{DD}^2$).
  - But this has the penalty of the delay in the circuits



Component (all in 2µm)	# of transistors	Area	Comments
Microcoded DSP Chip [21]	44802	94mm^2	20-bit datapath
Multiplier	20432	12.2mm^2	24x24 bits
Adder	256	0.083mm^2	conventional static
Ring Oscillator	102	0.055mm^2	51-stages
Clock Generator	56	0.04mm^2	cross-coupled NOR

Power consumption sources in CMOS circuits

- Power-delay product improves as delay increases: reduction of the power supply and operations at lower speed: Overall system throughput is maintained.
- Delay and energy behaviour as a function of VDD is 'well-behaved' and relatively independent of logic style and circuit complexity



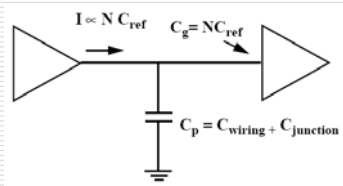
Power consumption sources in CMOS circuits

- Optimal transistor sizing with voltage scaling.
  - Transistor sizing in power consumption.

For low power it is important to equalise all delay paths, so that a single critical path does not unnecessarily limit the performance of the entire circuit.

Next, the W/L ratios should be uniformly raised for all devices, yielding a uniform decrease in the gate delay and hence allowing for corresponding reduction in voltage and power.

The drive of the first gate have to take into consideration the capacitance of the following stage plus the parasitic capacitance of the connection. The input capacitance of both stages is assumed to be $N \cdot C_{ref}$, C_{ref} = MOS device gate capacitance with smallest W/L



Delay through the first gate:

$$T_N = K\frac{C_p + NC_{ref}}{NC_{ref}}\frac{V_{ref}}{(V_{ref} - V_t)^2}$$

$$= K\left(1 + \alpha/N\right)\frac{V_{ref}}{(V_{ref} - V_t)^2}$$

Power consumption sources in CMOS circuits

- The evaluation of the energy performance of the two designs at the same speed (the delay remains constant and scales to $1/V_{DD}$) the voltage of the scaled version is

$$V_N = \frac{1 + \alpha/N}{1 + \alpha} V_{ref}$$

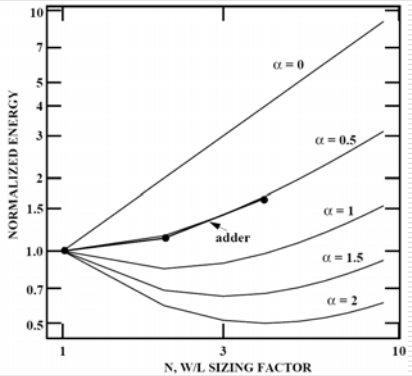
with N being the transistor size of the speed up circuit in front of transistor with unity size.

- Then the energy consumed by the first stage

$$\text{Energy}(N) = \left(C_p + NC_{ref}\right)V_N^2 = \frac{NC_{ref}\left(1 + \alpha/N\right)^3 V_{ref}^2}{(1 + \alpha)^2}$$

- Analysis of the expression tells that
  - The lowest power case occurs at $\alpha = 0$ → without parasitic capacitance
  - At high values of α (significant interconnection capacitances) there is an optimum value for N
- It results that the determination of an optimum supply voltage is a key to minimise the power consumption

Power consumption sources in CMOS circuits



Energy vs. transistor sizing factor

Energy(N)/Energy(1) vr N for various α factors

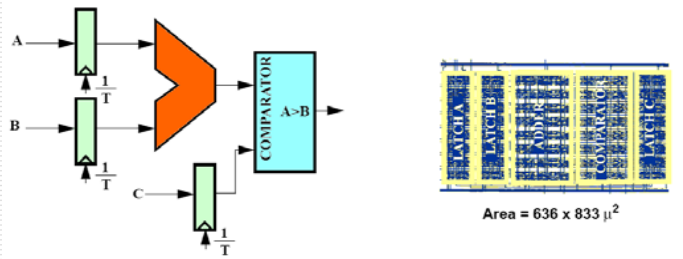
It results that exists an optimum sypply voltage that minimises power consumption.

- In submicron technologies, constant voltage scaling results in higher electric fields that create hot carriers, with device degradation in time.
- Hot carrier creation can be reduced using lightly doped drain devices. In this case, an optimal 2.5V was found for a 0.25μm technology by choosing the minimum point on the delay vs. V_{DD} curve

Power consumption sources in CMOS circuits

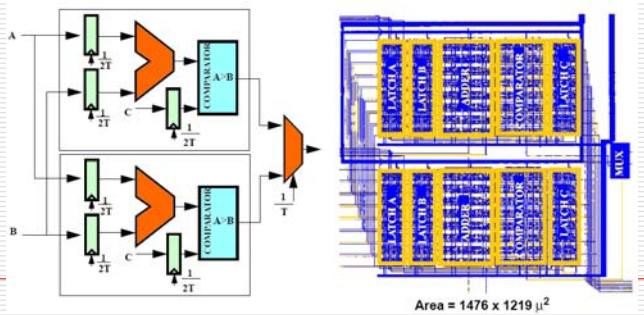
- Architecture driven voltage scaling in an 8-bit datapath.
  - CMOS logic gates achieve lower power-delay products as the supply voltages reduce → Speed has to be compensated.
  - A case to study: An 8-bit data-path consisting of an adder and a comparator is analysed assuming a 2.0μm technology
  - Reference → Worst case: delay = 25ns at V_{DD}=5V

$P_{ref} = C_{ref} \cdot V_{ref}^2 \cdot f_{ref}$, C_{ref} = effective capacitance being switched per cycle



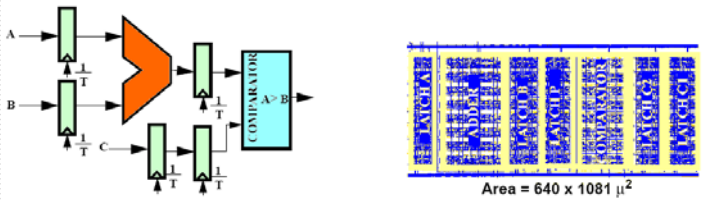
Power consumption sources in CMOS circuits

- Solution 1. Parallel implementation of the datapath
- Maintain the throughtput while reducing the supply voltage .
  - Each unit works at half the original rate maintaining the original throughput: from 25ns to 50ns of clock period).
  - Voltage can be dropped to 2.9V
  - Datapath capacitance increase to 2.15 (a slightly more than the expected 2 factor; clock reduces at a factor of 2).
  - Then $P_{par} = C_{par} \cdot V_{par}^2 \cdot f_{par} = (2.15\ C_{ref})(0.58\ V_{ref})^2(f_{ref}/2) \approx 0.36\ P_{ref}$.
- Cost in parallelism:increase in area.



Power consumption sources in CMOS circuits

- Solution 2. Pipelining implementation of the datapath
- Only an additional latch.
  - Critical path: the max[T_{adder}, $T_{comparator}$], allowing the adder and the comparator to operate at a slower rate
  - Voltage can be dropped to 2.9V again
  - Then $P_{pip} = C_{pip} \cdot V_{pip}^2 \cdot f_{pip} = (1.15\ C_{ref})(0.58\ V_{ref})^2(f_{ref}) \approx 0.39\ P_{ref}$.
- Lesser increase in area.



- Solution 1+2. Parallelism + pipelining
- Critical path (speed requirement) reduced by a factor of 4 .
  - $P_{parpip} = C_{parpip} \cdot V_{parpip}^2 \cdot f_{parpip} = (2.5\ C_{ref})(0.4\ V_{ref})^2(f_{ref}) \approx 0.2\ P_{ref}$

Optimisation techniques

- Levels of abstraction from system to circuit design
  - Circuit description is transformed and manipulated at different levels of abstraction:
    - System level. System described in terms of software, hardware and memory components with algorithms that perform a certain functionality.
    - Behavioral or architectural level. Individual components described in terms of their algorithmic behavior. Descriptions usually build using specific languages like VHDL, Verilog, or general purpose as C.
    - Register transfer level. Hardware described in terms of arithmetic modules, registers, multiplexors and interconnect to steer data flow.
    - At gate level functionality of the circuit is described in terms of netlist or a set of boolean equations.
    - At the transistor level the circuit is described in terms of their network structure.
    - On the physically level the circuit is described in terms of the mask layout to be fabricated
- Partitions of the system in components have great incidences in terms of power consumption → Power have to be taken in consideration as soon as possible in the design flow.

Optimisation techniques

- Power consumption optimisation at the system level
  - Algorithm selection has to be made for best meeting design constraints. Power consumption of an algorithm depends on its characteristics (overall complexity and basic operations complexity).
    - Power consumption due to capacitance switching at the algorithmic level can be reduced if the computation task could be performed with fewer operations:
      - Example: vector quantization technique of a lossy compression technique, for coding video data: for a vector size of 16, the distorsion metric calculation involves 16 memory accesses, 16 substractions, 16 multiplicacions and 16 additions.
      - In order to minimize a distorsion metric, a codeword for each input is chosen.
      - Election of algorithms for implementation:

	#memory accesses	#multiplications	#adds	#subs
Full search	4096	4096	3840	4096
Tree search	256	256	240	264
Differential tree search	136	128	128	0

Optimisation techniques

- Memory access and management. Data transfers to memory are a lot more power consuming than word multiplication.
- Hardware/software partitioning. At high level, decisions on where to execute a process in hardware or software have to be made, in terms of flexibility, timing, performance, and power consumption.

  Tiwari at alt. showed that also power consumption of software can be optimised.

  Partitioning is important in order to minimise the number of off-chip operations, since off-chip operations imply a significant amount of power consumption.

  Many chips that use system-on-chip processors are nowadays available as synthesizable cores.
- Quiescent unit shutdown are used at different levels of abstraction. Mutually exclusive processes allow the clock shutdown of the process while it is not operating.
- Voltage scaling. Power consumption depends on the square of the power supply. Lowering supply voltage is an efficient means to lower power consumtion.

  Throughput of the circuit can be architecturaly compensated by means of area incresing: parallel implementation and pipelining

Optimisation techniques

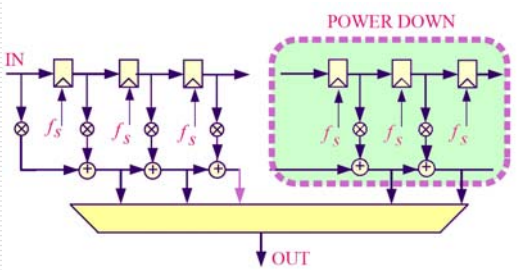
- Power consumption optimisation at the behavioural level
  - Algorithm transformation. Slow operations replaced by faster ones.
    - For example, the multiplication by constants by shift and add operations.
    - Prepare behavioral description for latter efficient power optimisations transformations.
  - Clock scheduling. Imposing a maximum delay in a clock cycle helps reducing power operation in a system.
  - Eliminating redundant computation.

    Winograd introduces incremental refinement structures for signal processing transformations with control strategies for low power.

    It is applied to FIR filter design. The number of filter taps used is dynamically varied to provide stop-band attenuation in proportion to a simple estimate of the time-varying energy in the undesired components of the input signal.

    The approach lowers the number of taps used to produce each output sample in correspondance to the processign task to be performed. That is, controlling the number of taps that must switch on proportionally to the required stopband, power savings can be achieved

Optimisation techniques



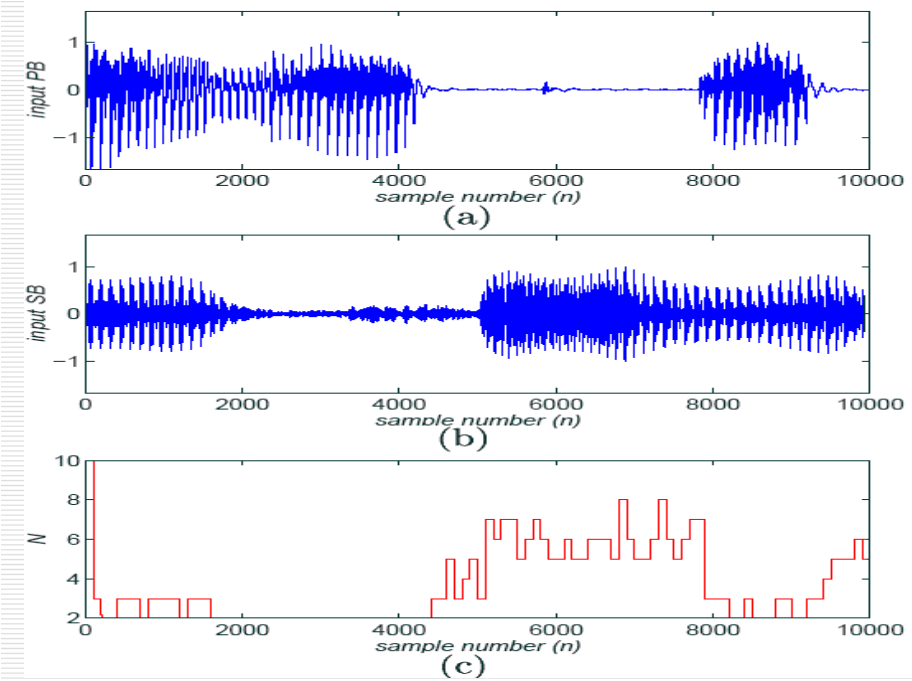
The incremental refinement structure along with an adaptation strategy was applied to two speech signals which had been frequency-division multiplexed: one signal was in the passband region of the lowpass filter and the other in the stopband region. The samplig rate for the FDM speech was 16 KHz.

Figure shows the FDM speech demultiplexing using low-power frequency selective filtering

a) Corresponds to the speech signal in the passband region

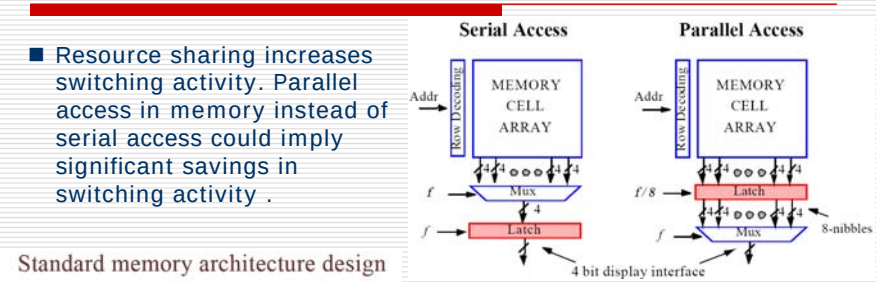
b) It is the speech signal in the stopband region.

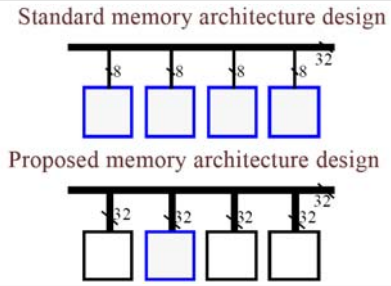
c) Number of filter sections used by the adaptive filtering technique.



Optimisation techniques

- Resource sharing increases switching activity. Parallel access in memory instead of serial access could imply significant savings in switching activity .

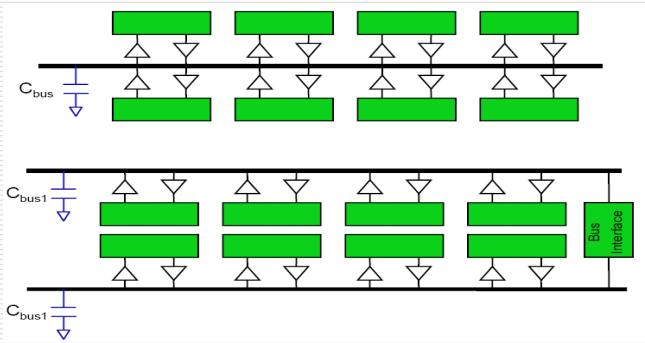




- Power consumption optimization for memory access. Only the block to be accessed is decoded. It reduces power by x4.

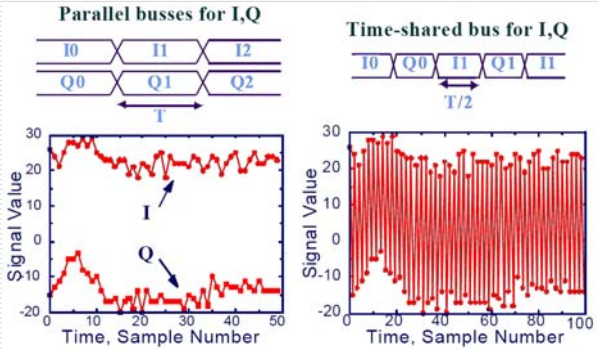
Optimisation techniques

- Single bus connected to all modules results in a large capacitance due to
  - The large number of drivers and receivers sharing the same bus
  - The parasitic capacitance of the long bus line
- A segmented bus structure will increase overall routing area but will significantly reduce the switched capacitance



Optimisation techniques

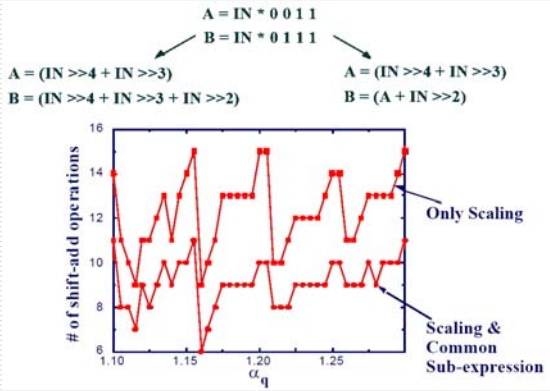
- Resource sharing in the execution of different operations on the same module in different cycles has a significant impact on power reduction, since it determines the switching activities in the module
- Time multiplexed architectures can destroy signal correlations and increase signal activity



Optimisation techniques

- Resource selection. Selection of the functional units in which operations are implemented. Pre-execution data analysis can help low power design

  Example: Optimising multiplications



Optimisation techniques

- Number representation in bus encoding.

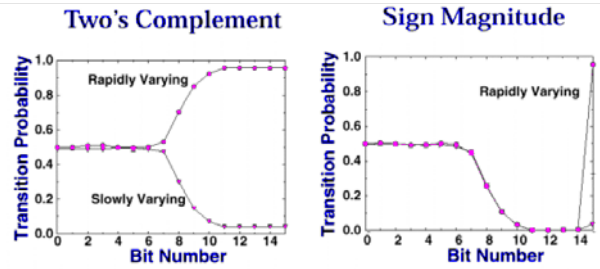
  In data words highly correlated, the sign magnitude representation have lower power consumption than the classical 2's complement representation.

  Alternatives to binary codes are

  - One-hot coding
    - A memoryless redundant code with rate N to 2^N
    - Every codeword has exactly one bit equal to 1
    - Good performance in reducing power consumption
    - Only realisable for small values of N
  - Gray codes
    - Memoryless non-redundant codes (rate=1) with Hamming distance equal to 1
    - Good for sequential coding with highly correlated data
    - Increment/decrement counters can be implemented directly as Gray counter
    - Binary codes are converted to Gray codes using XOR gates
  - Bus inversion codes
    - Redundant coding with rate N to N+1
    - Memory coded to keep the number of bit transitions from the previous encoded value minimal: n input bits are either inverted or unchanged
    - An extra bit (memoryless) indicates to the decoder if the other N bits need to be inverted or not
    - Good reduction of the switching activity, specially for small values of N

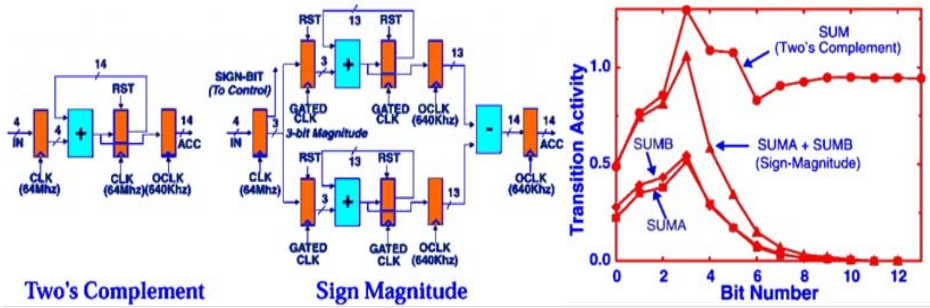
Optimisation techniques

- Sign-magnitude representation
  - Numerical representation that uses exactly one bit for the sign and the others represent the absolute value of the number
  - 2's complement representation has the advantage of making arithmetical operations easy to implement, but the sign-bit extension brings a lot of bit toggling activity if low amplitude signals swings frequentely around to zero
  - So, the advantage of sign magnitude with respect to 2's complement depend on correlation and amplitude of the signal to be represented



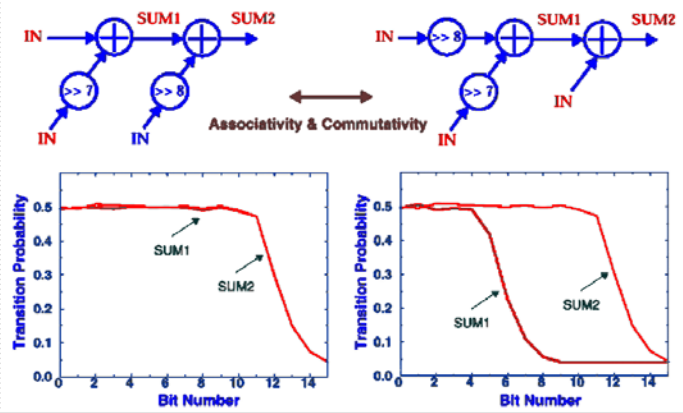
Optimisation techniques

- So, different architectures performing the same arithmetical or logical operation can have different switching activity in the intermediate nodes of the datapath.
- Using alternative data representations and reordering the inputs are two techniques that can lead to reduced switching activity. The characteristics of the signals to be processed should be considered in order to choose the best architecture. In the next example the accumulator sign magnitude datapath switches 30% less capacitance for uniformly distributed inputs.



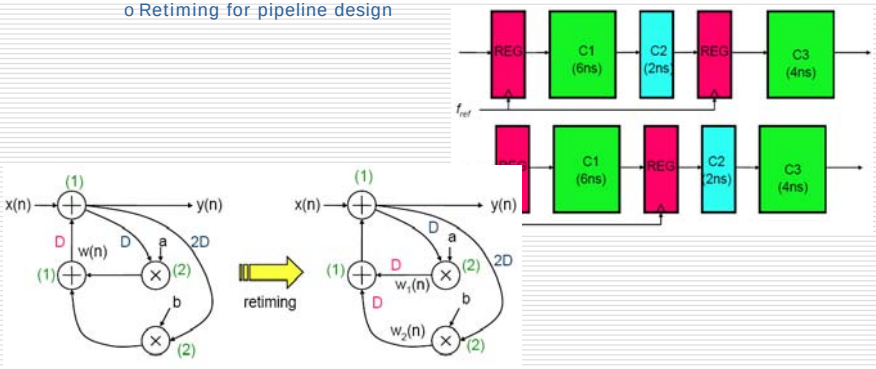
Optimisation techniques

- Reordering inputs
  - The mathematical properties of addition are used to reduce the switching activities in the intermediate result.

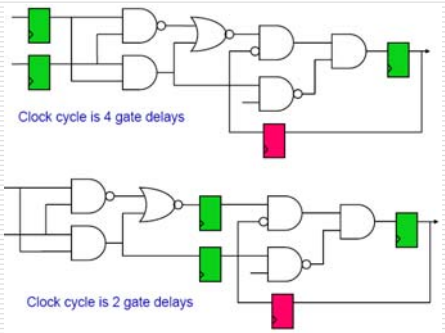


Optimisation techniques

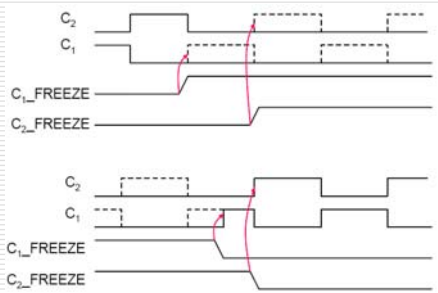
- Reduction in control units can also be achieved selecting 1-code Hamming distances codification between neighbouring states. This can be achieved using the known Gray codes or one-hot codes
- Retiming. Is a transformation technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit
  - Retiming for pipeline design



Optimisation techniques



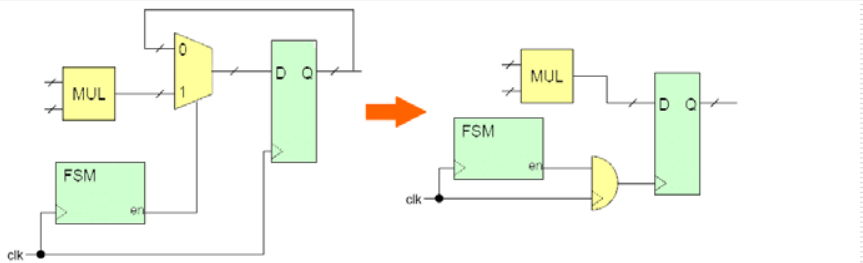
- Retiming in registered flip-flops



Optimisation techniques

- Power consumption optimisation at RTL level
  - At RTL level automatic CAD tools can help in the transformation to register level netlists for optimum power consumption.

    Here are resumed some of the most used techniques:
    - Clock gating. The clock of sequential modules (registers, …) are turned down when the module performs no operation. The final effect is a reduction in the fanin of the register data input and a reduction in the overall capacitance of the clock signal .



Optimisation techniques

- Operand isolation. A similar technique to clock gating is applied to input data in combinational blocks.

Simple Decoder

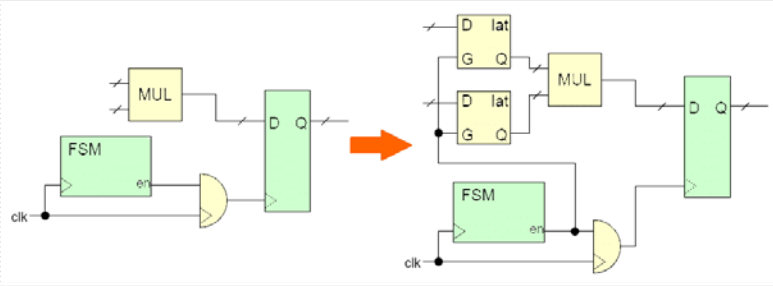
```
module decoder (a, sel);
    input  [1:0[ a;
    ouput [3:0] sel;
    reg    [3:0] sel;
    always @(a) begin
        case (a)
            2'b00: sel=4'b0001;
            2'b01: sel=4'b0010;
            2'b10: sel=4'b0100;
            2'b11: sel=4'b1000;
        endcase
    end
endmodule
```

Decoder with enable

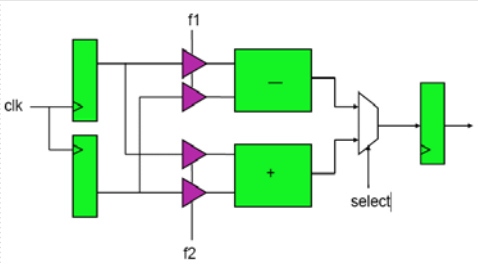
```
module decoder (en,a, sel);
    input          en;
    input  [1:0[ a;
    ouput [3:0] sel;
    reg    [3:0] sel;
    always @({en,a}) begin
        case ({en,a})
            3'b100: sel=4'b0001;
            3'b101: sel=4'b0010;
            3'b110: sel=4'b0100;
            3'b111: sel=4'b1000;
            default: sel=4'b0000;
        endcase
    end
endmodule
```


Optimisation techniques

In this case, a multiplexor controls when an operand arrives to the input of a combinational block for operation, avoiding reduntant operation in the block and, so, preventing switching activity propagation. Operand isolation was presented (Correale 1995) as one of the techniques for power consumption reduction in PowerPCxx family of embedded cores .



Optimisation techniques

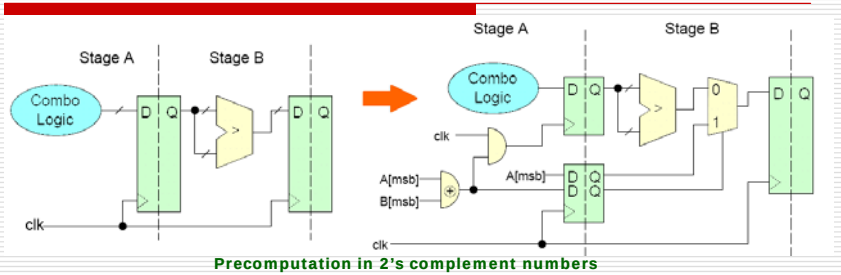


Preventing switching activity by path selection

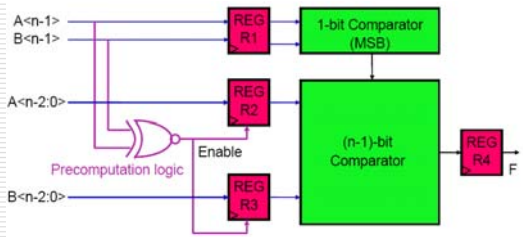
- Precomputation is used to reduce power consumption in individual stages of pipelined designs. It is useful to have specific signals that prevents further processing when results can be predicted. Gating of combinatorial logic by this signals prevents further switching and reduces power consumption.

  For exemple, in the comparison of 2's complement numbers, sign bit can be used to precompute the result. Whenever MSB differs, logic can be added to pass the positive number, avoiding switching activity.

Optimisation techniques



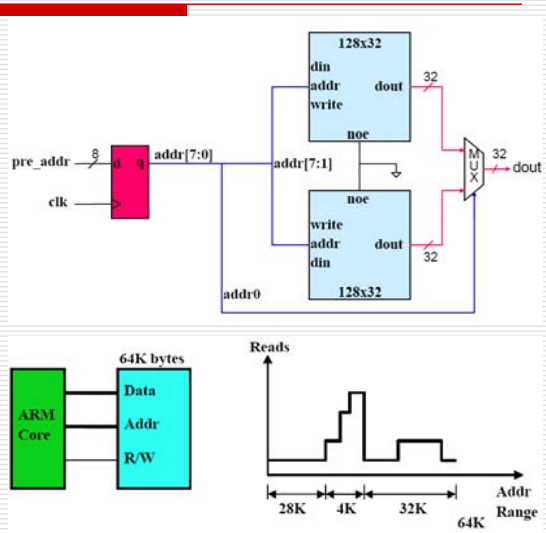
Precomputation in 2's complement numbers



Number's comparison by precomputation

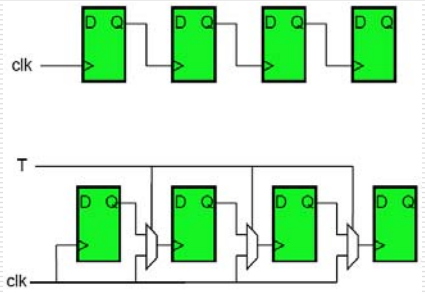
Optimisation techniques

- Memory partition. Application driven logic can significantly reduce witching activity .

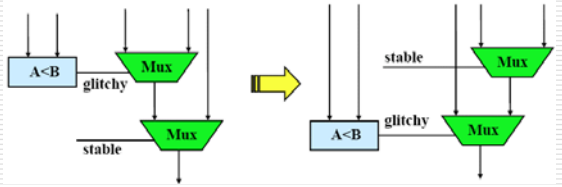


Optimisation techniques

- Keep on DFT rules .



- Reordering of signals can also reduce switching activity

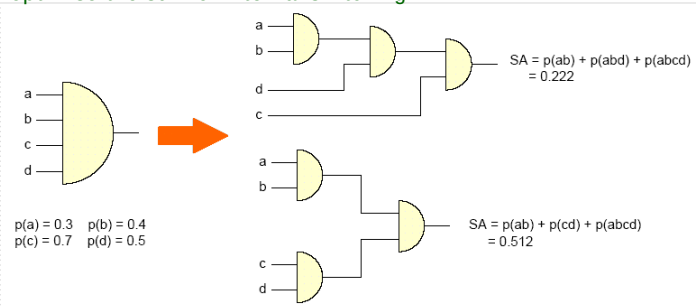


Optimisation techniques

- Gate level
  - Most commercial available tools offer support for low power optimisation when mapping from RTL to gate level.

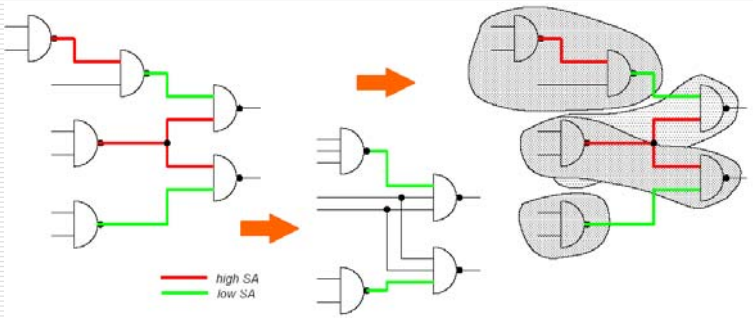
    Usually process is divided into technology decomposition and technology mapping.

    Technology decomposition is the process that transforms a logic network into a known set of basic gates. For low power it usually optimise the sum of internal switching .



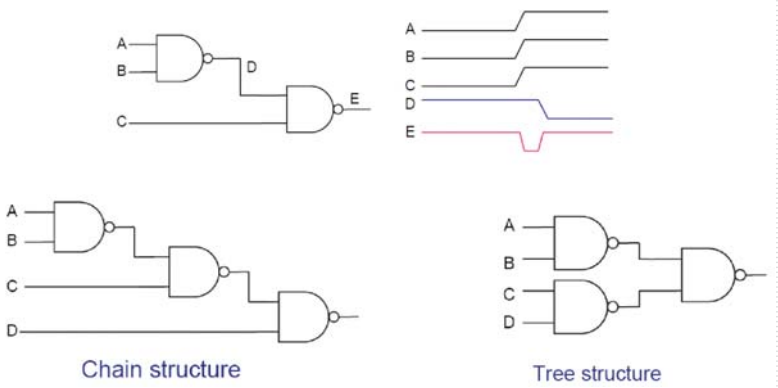
Optimisation techniques

- Technology mapping is the implementation of a technology decomposition from the actual cell library. The objective of the decomposition is to minimise the total power dissipation by reducing the total switching activity. Usually, low power technology mapping can hide highly active signals within gates .



Optimisation techniques

- Minimising glitches. Glitches are spurious transitions due to unbalanced delay paths. Low power design trades minimizing unbalanced paths. In this sense, tree-architectures have fewer glitches than chain-architectures .



Optimisation techniques

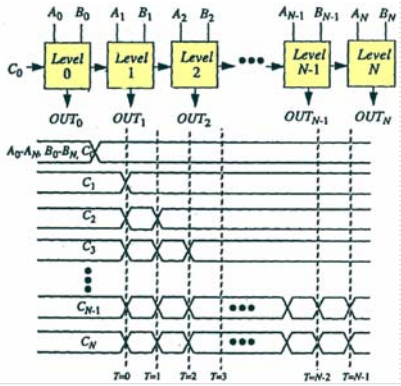
- Transistor and physical level
  - In the physical level power consumption can be addressed by a lot of different circuit techniques. For logic and sequential elements there exist many techniques. Power consumption can be addressed minimising leakage power, switching power, capacitance between highly active signals, minimisation of transition times, …
  - Between the possibilities offered at transistor level, perhaps are the adiabatic circuits which most new concept introduced in power consumption reduction.

    Adiabatic circuits offer the possibility to recycle the energy drawn from power supply .

Circuit design and technology considerations

- Dynamic vs static logic
  - Dynamic logic appears to have better advantages (including reduced switching activity due to hazards) for low power performances. Static logic, on the other side, has no precharge operation and charge-sharing does not exist.
  - Several considerations must be made
    - Spurious transitions. Static designs can exhibit spurious transitions (called races and dynamic hazards) from one logic block to the next

      For example: in an N-bit adder, the summation of 11…11 + 11…11 + 0 (carry in) gives as a final result all zeros, but a spurious one have been propagated dynamically from the lowest significant bit to the most significant bit. In terms of power this can imply an increase in energy of about 30%.



Circuit design and technology considerations

In fact, the number of extra transitions (or glitching transitions) is a function of the logic depth, the signal skew due to different arrival time of the input and the signal pattern. Of course, the worst case occurs when the extra transitions are a factor $O(N^2)$, being N the logic depth.

The dinamic component of switching power is easily analysed in this case, where the output will transition N times in the worst case, being the extra transitions for all the N stages of . In reality, the transition activity due to glitching will be less since the worst input pattern will occur infrequently.

- Dynamic logic has the disadvantage to need precharge operation. Since in dynamic logic every node must be precharged every clock, this means that some nodes are precharged and, immediately, discharged again, as the node is evaluated. This leads to a higher activity factor