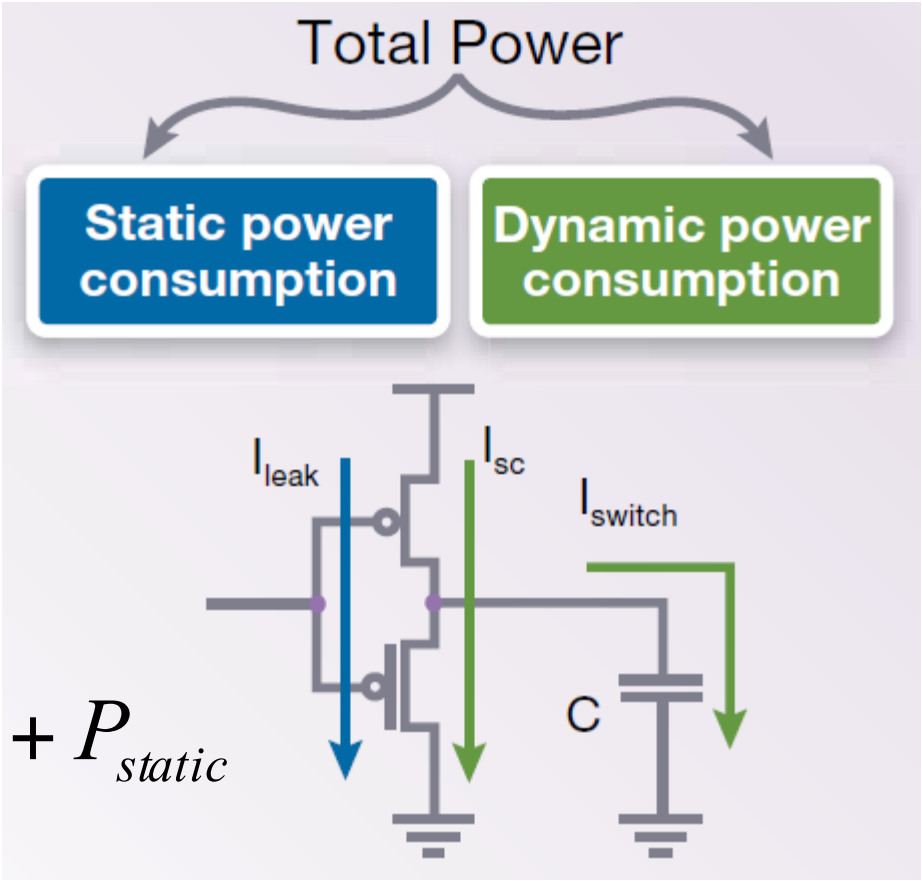
Lecture 20
Power Optimization (Part 1)

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese461/

Power Dissipation

- Dynamic power consumption
  - switching current
- Static power consumption
  - short-circuit current
  - leakage current



$$P_{avg} = P_{dyn} + P_{short} + P_{lkg} + P_{static}$$

Dynamic Power

- Switching current
  - energy dissipated in half a cycle

$$\int_0^{1/(2f)} CV\left(\frac{dV}{dt}\right)dt = \int_0^{V_{DD}} CVdV = \frac{1}{2}CV_{DD}^2$$

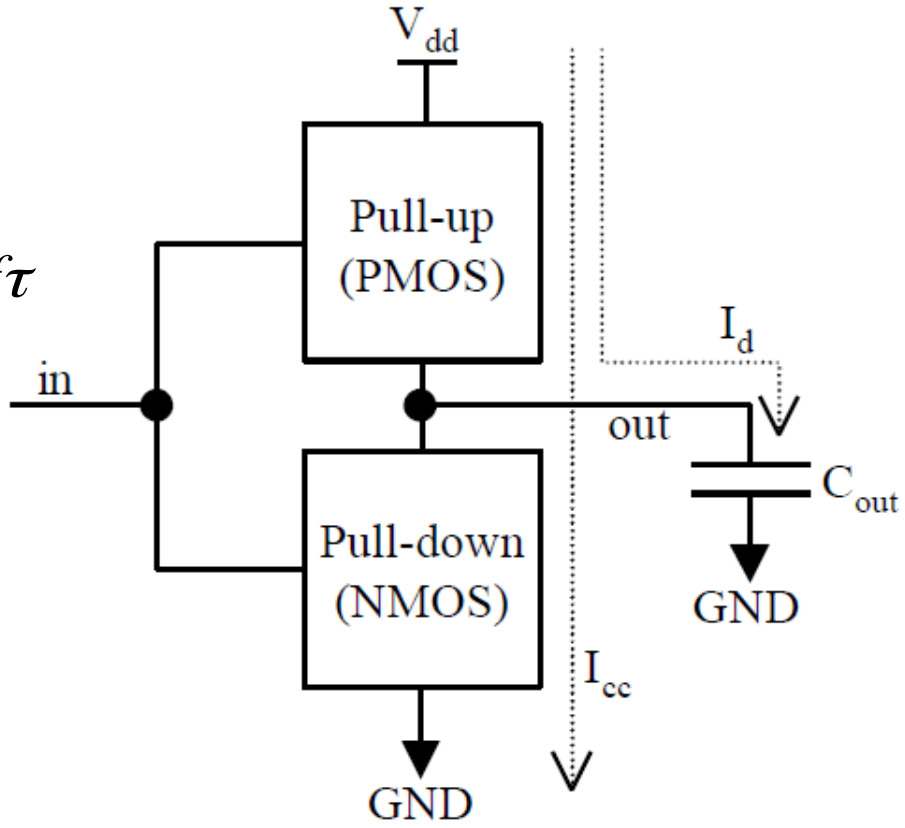
  - dynamic power expression
  - α is the average number of rising transitions in one cycle

$$P_{dyn} = \alpha C_{out} V_{DD}^2 f$$

Static Power

- Short-circuit currents
  - ß is the gain factor of s MOSFET
  - V_{th} is the threshold voltage
  - τ is the rise/fall time

$$P_{short} = \alpha \frac{\beta}{12}(V_{DD} - 2V_{th})^3 f\tau$$



Static Power

- Leakage currents

$$P_{lkg} = (I_{diode} + I_{subthreshold}) \cdot V_{DD}$$

- Other static power
  - current flow from VDD to GND during idle time
  - historically, NMOS circuits has high static power
  - CMOS static power should be 0
  - might result from bus conflict where multiple drivers attempt to drive a signal to different logic values

Low Power Design Methodologies

- Adapt process technology
  - reduce capacitance
  - C_{fo} is the input capacitance of fan-out gates
  - C_w is the wire capacitance
  - C_p is the parasitic capacitance

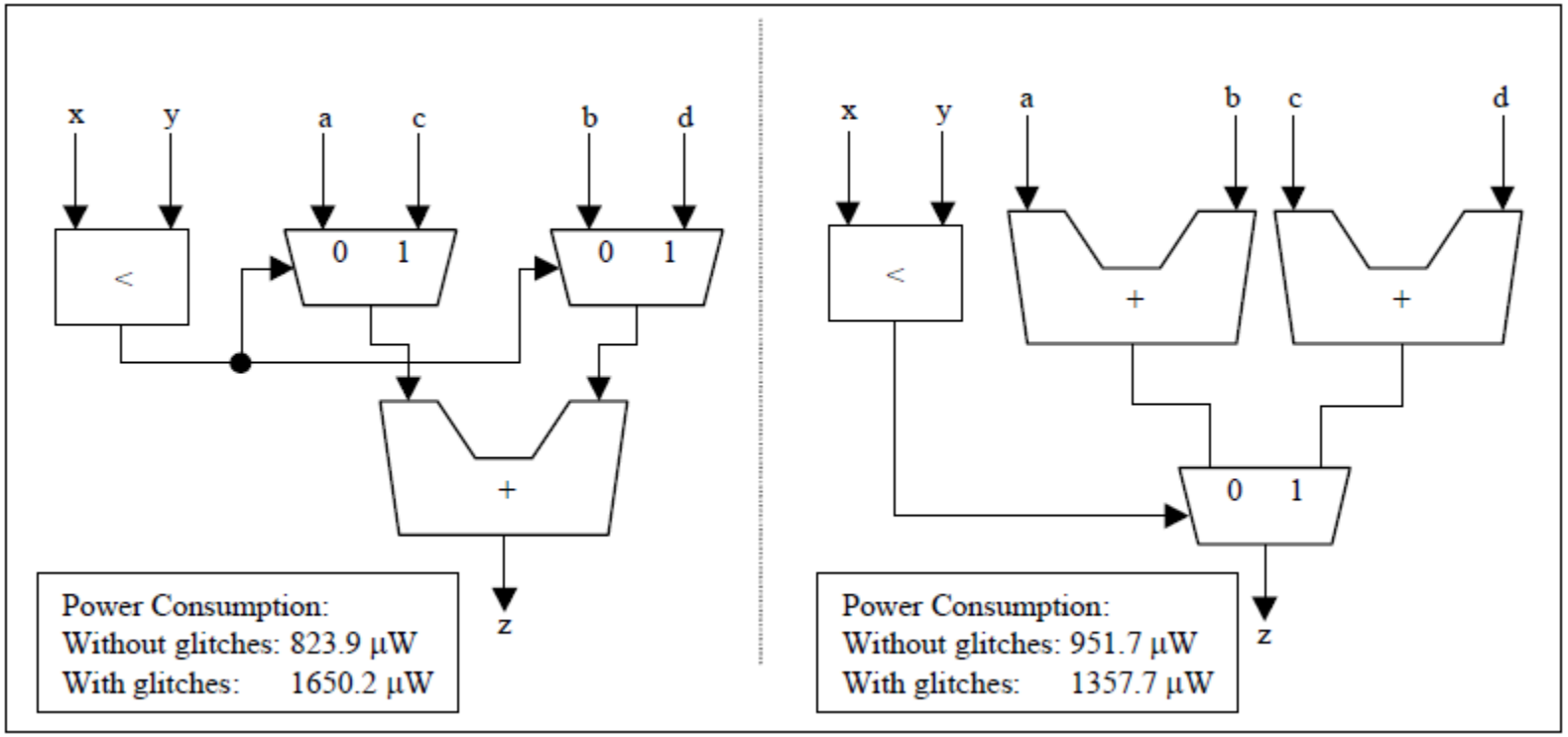
$$C_{out} = C_{fo} + C_w + C_p$$

  - reduce leakage current
  - reduce supply voltage

$$T_d = \frac{C_{out} V_{DD}}{I} = \frac{C_{out} V_{DD}}{\eta \left(\frac{W}{L} \right) \left(V_{DD} - V_{th} \right)^2}$$

Low Power Design Methodologies

- Reduce switch activity
  - minimize glitches



  - nodes logically deeper more prone to signal glitches

Low Power Design Methodologies

- Reduce switch activity
  - minimize glitches
  - minimize number of operations
  - example: vector quantization (VQ) algorithm
  - X_i are the elements of the input vector
  - C_{ij} are the elements of the codebook vector

$$D_i = \sum_{j=0}^{15} \left(X_j - C_{ij} \right)^2$$

Algorithm	# of Memory Access	# of Multiplications	# of Adds	# of Subs
Full Search	4096	4096	3840	4096
Tree Search	256	256	240	264
Differential Tree Search	136	128	128	0

Low Power Design Methodologies

- Reduce switch activity
  - minimize glitches
  - minimize number of operations
  - low power bus

- Examples
  - one-hot coding
  - gray coding
  - bus-inversion coding

$$(B(t), INV(t)) = \begin{cases} (b(t), 0) & \text{if } H <= N/2 \\ (b'(t), 1) & \text{Otherwise} \end{cases}$$

N: number of bus lines, H: Hamming Distance

Minimize Data Transition on Bus

```
// Code that resets the bus to default
status after valid gets de-asserted

always@(posedge clk or negedge reset)

begin
   if(!reset)
   data_bus <= 16'b0    ;
   else if (data_bus_valid)
   data_bus <=  data_o ;
   else
   data_bus  <= 16'b0   ;
end
```

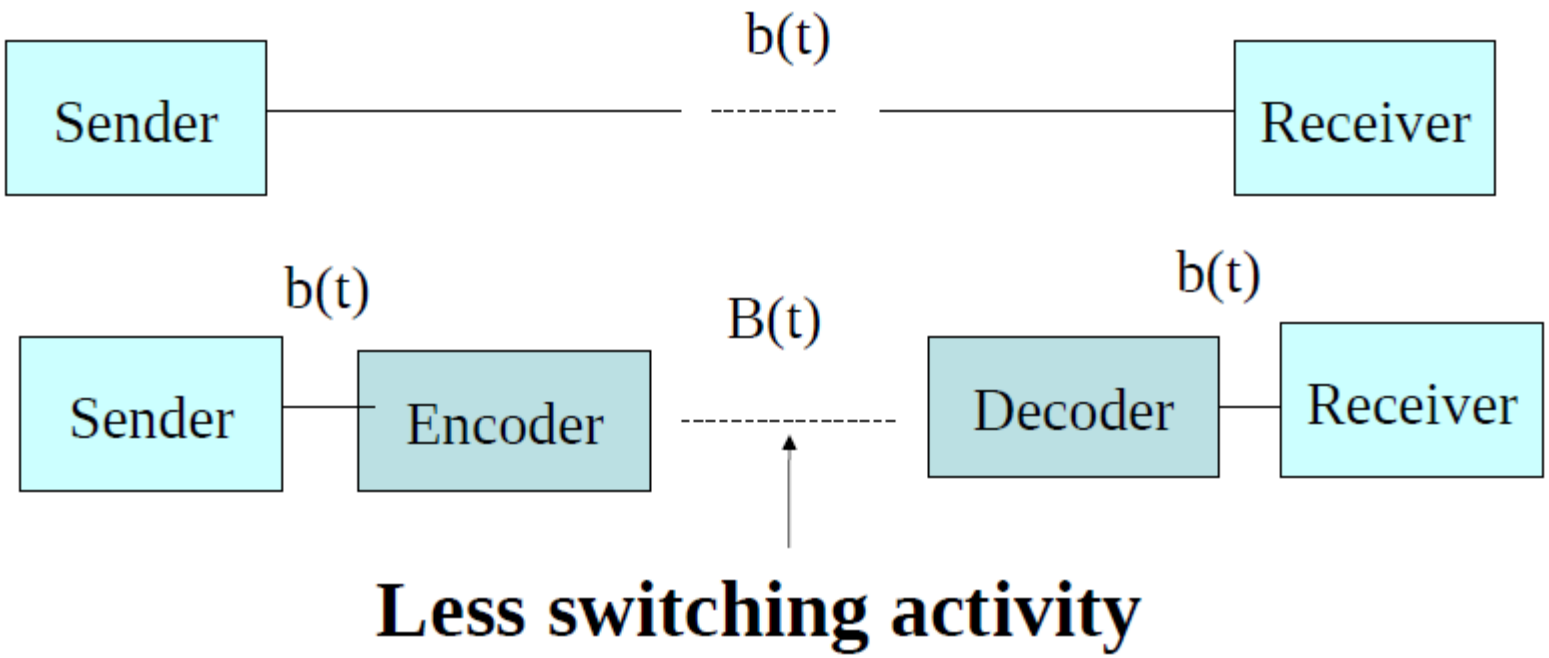
```
// Code that holds the bus to its previous
value after valid gets de-asserted

always@(posedge clk or negedge reset)

begin
   if(!reset)
   data_bus <= 16'b0    ;
   else if (data_bus_valid)
   data_bus <=  data_o ;
end
```

Bus Width	Total random data input	Number of Transitions without coding (A)	Number of transition with bus invert coding (B)	Percentage improvement in bus invert coding against non-coded data (B) / (A) * 100
32-bit	**5000000**	**8005314**	**6879054**	14.06 %
64-bit	1000000	32000980	28513273	**10.89**

Bus Coding



Less switching activity

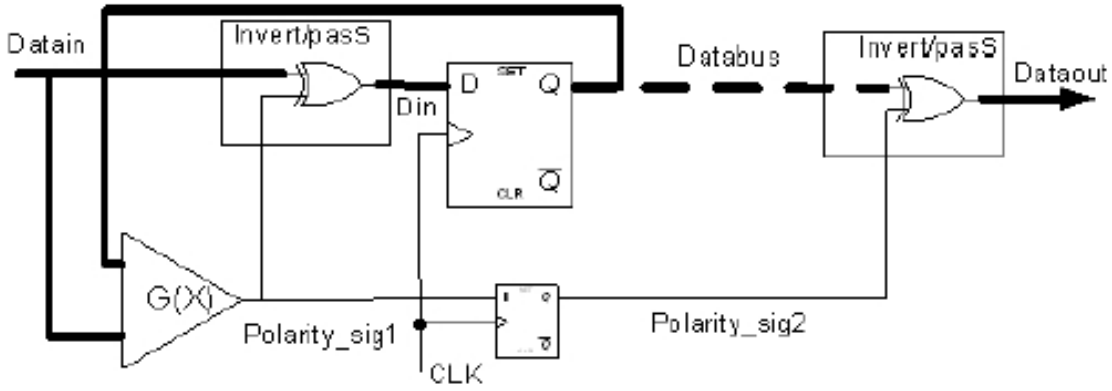
b(t): Source word

B(t): Code word

Bus Invert Coding

Binary (31 Trs)	
00101010	
00111011	2
11010100	7
11110100	1
00001101	6
01110110	6
00010001	5
10000100	4

BIC (19 Trs)		
00101010		
00111011	0	2
00101011	1	2
00001011	1	1
00001101	0	3
10001001	1	3
00010001	0	4
10000100	0	4



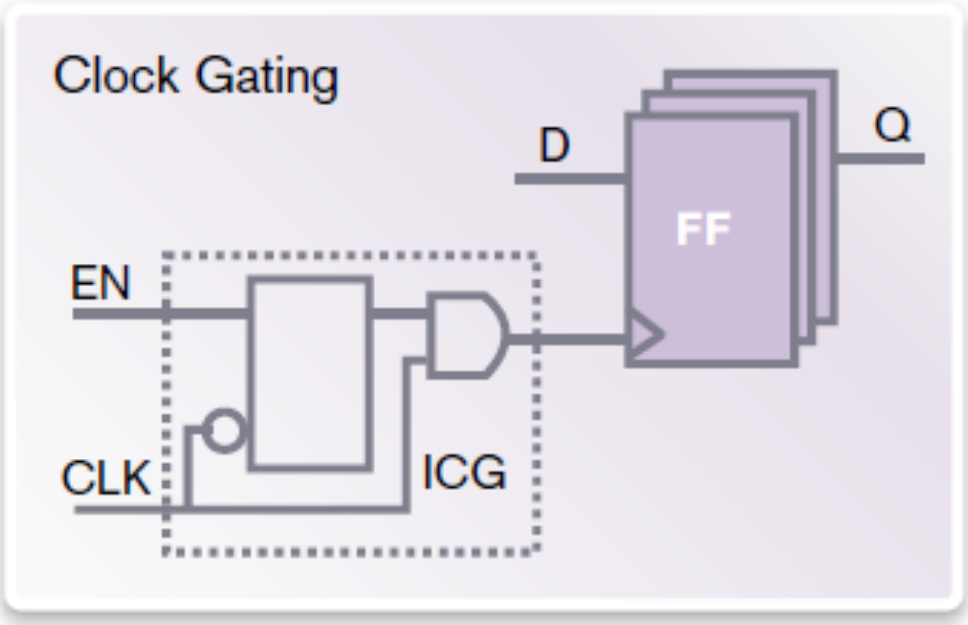
Low Power Design Methodologies

- Reduce switch activity
  - minimize glitches
  - minimize number of operations
  - low power bus (state machine encoding)
  - scheduling and binding optimization

control-data-flow graph (CDFG)

Low Power Design Methodologies

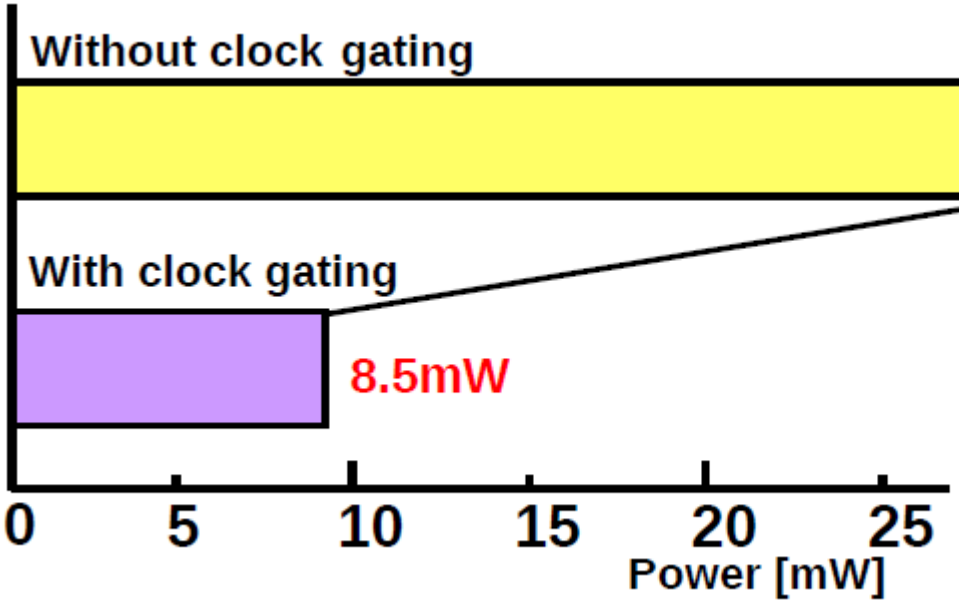
- Power down modes
  - clock gating

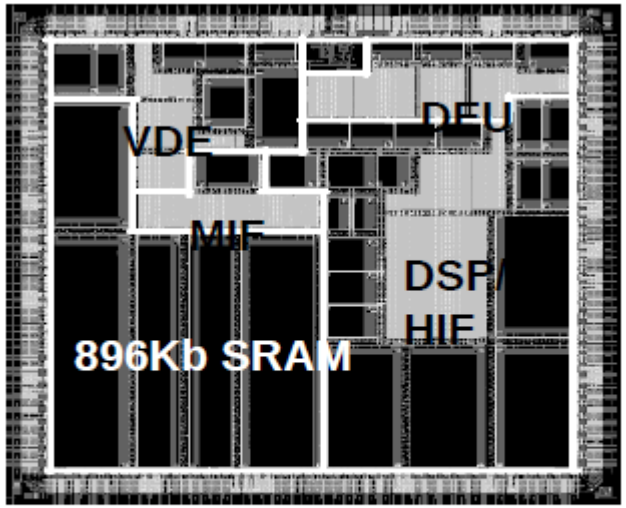


  - enabled flip-flops
  - memory partitioning
  - power gating

How Effective is Clock-Gating?

- 90% FF clock-gated
- 70% power reduction

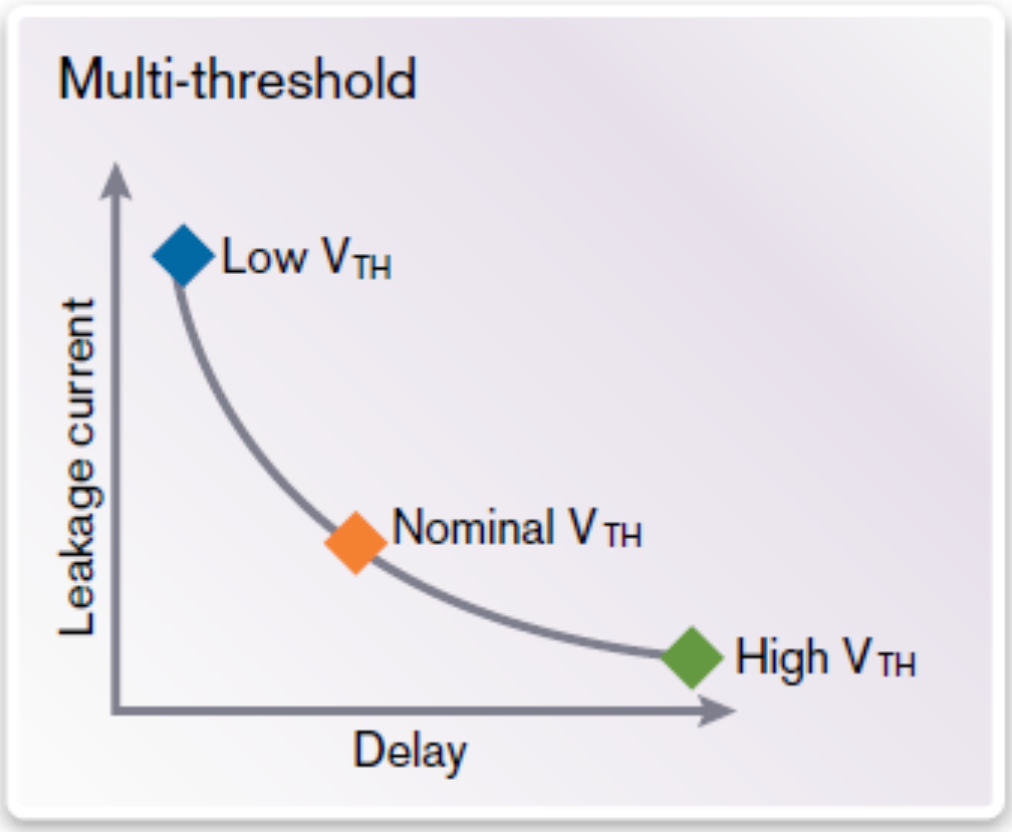




MPEG4 decoder

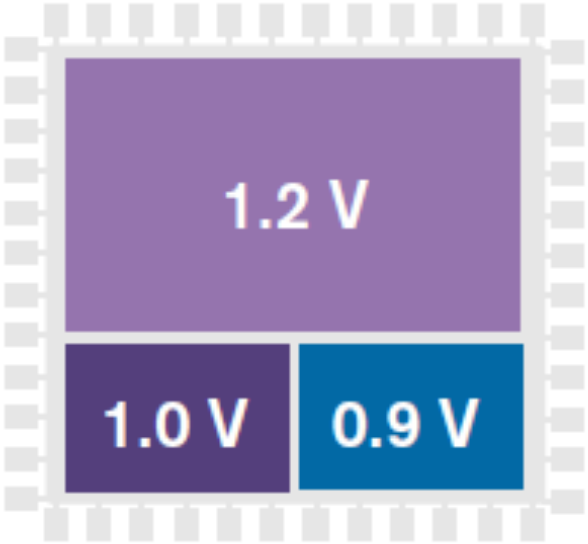
Low Power Design Methodologies

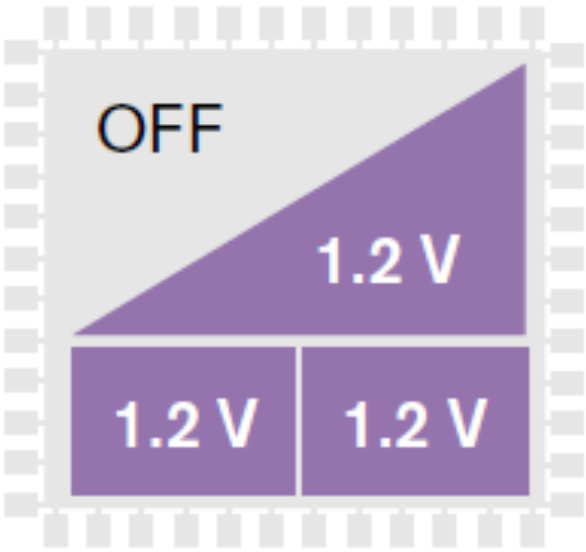
- Voltage optimization and scaling
  - multi-Vth optimization



Low Power Design Methodologies

- Voltage optimization and scaling
  - multi-voltage domain
  - dynamic voltage frequency scaling





Resource Sharing

- Save area and power

```
// Example where resource sharing is not
possible

always@(in1 or in2 or sel)
if(sel)
    out1 = in1 + in2 ;
else
    out1 = 4'b0 ;

always@(in3 or in4 or sel)
if (!sel)
   out2 = in3 + in4 ;
else
    out2 = 4'b0 ;
```

```
// Example where resource sharing is
possible

always@(in1 or in2 or sel or in3 or in4)
if(sel)
begin
    out1 = in1 + in2 ;
    out2 = 4'b0 ;
end
else
begin
    out1 = 4'b0 ;
    out2 = in3 + in4 ;
end
```

Questions?

Comments?

Discussion?