
DO NOT SHARE
SLIDES AND CLASS MATERIALS
ON ONLINE SITES
Course Hero

Advanced Logic Design
Low Power Logic Design

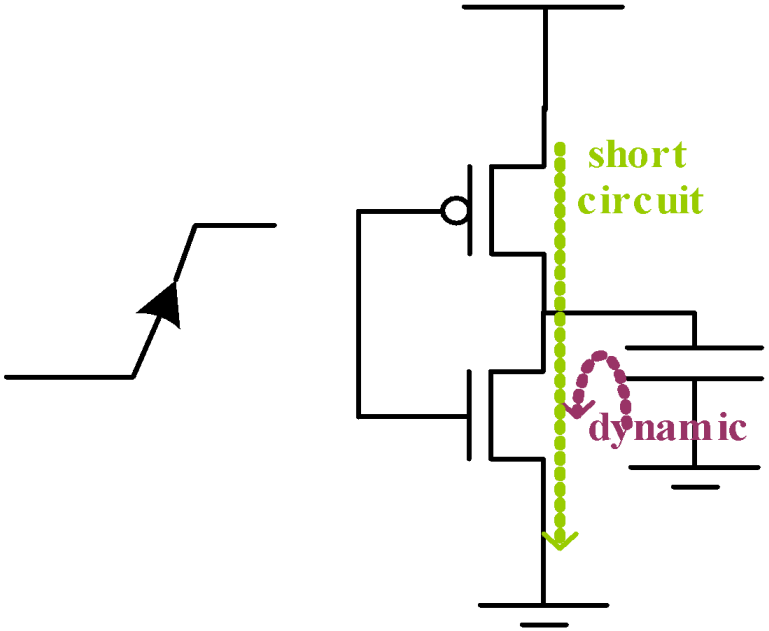
Mingoo Seok
Columbia University

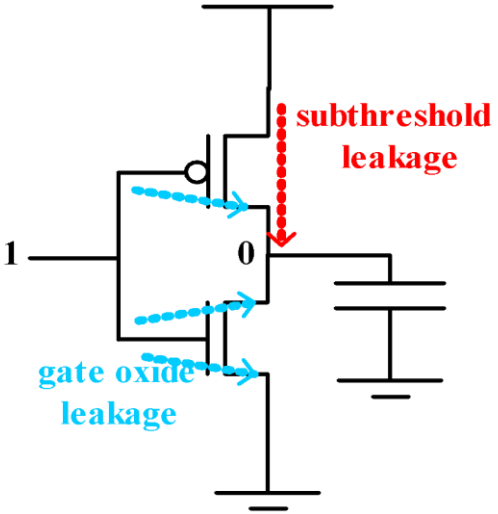
Slide sources: Digital Design by John F. Wakerly (1999)

Digital IC Power Consumption

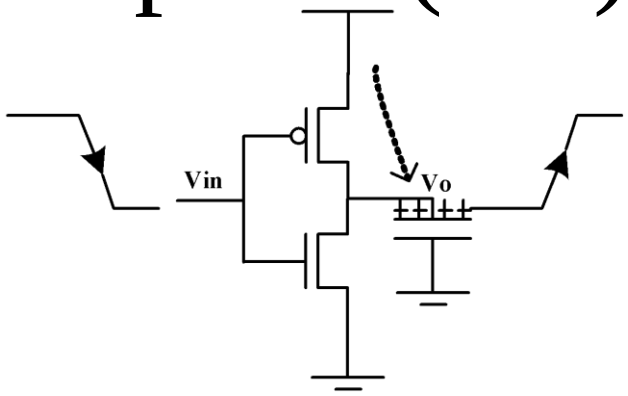
- Total Power =

  Dynamic Power + Static Power + Short Circuit Power





Dynamic Power Consumption (1/2)



- Input 1→0

–Energy drawn from power supply:

$$E_{supply} = \int_0^\infty P(t)\cdot dt = \int_0^\infty V_{dd} i(t)\cdot dt = \int_0^{V_{dd}} V_{dd} C \cdot \frac{dV_O}{dt}\cdot dt = V_{dd} C \cdot \int_0^{V_{dd}} dV_O = CV_{dd}^{\ 2}$$

– Energy consumed by PMOS:

$$E_{PMOS} = \int_0^\infty P(t)\cdot dt = \int_0^\infty (V_{dd} - V_O)\cdot i_p(t)\cdot dt = C\cdot \int_0^{V_{dd}} (V_{dd} - V_O)\cdot dV_O = \frac{1}{2} CV_{dd}^{\ 2}$$

– Energy in the capacitor:

$$E_{CAP} = \int_0^\infty V_O \cdot i_C(t)\cdot dt = C\cdot \int_0^\infty V_O \cdot \frac{dV_O}{dt}\cdot dt = C\cdot \int_0^{V_{dd}} V_O \cdot dV_O = \frac{1}{2} CV_{dd}^{\ 2}$$

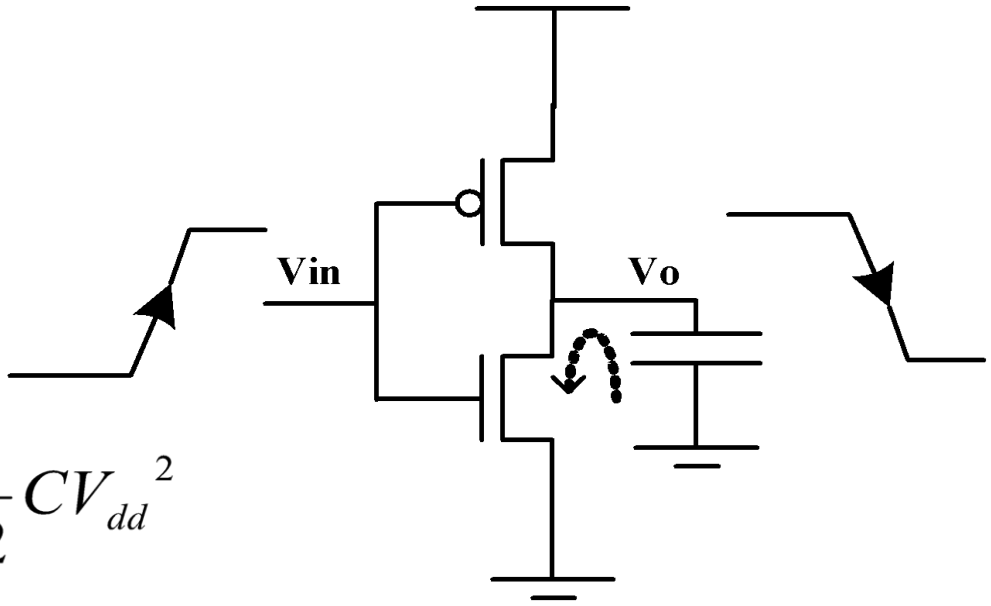
– Power:

$$\boxed{P = f\cdot E_{PMOS} = \frac{1}{2} fCV_{dd}^{\ 2}}$$

Dynamic Power Consumption (2/2)

- Input 0→1
  - Energy drawn from supply: 0
  - Energy consumed by NMOS equals to the energy stored on the capacitance:

$$E_{NMOS} = \int V_O i(t) \cdot dt = \frac{1}{2} C V_{dd}^{\ 2}$$



  - Power:

$$P = f \cdot E_{NMOS} = \frac{1}{2} f C V_{dd}^{\ 2}$$

Static Power Dissipation

- Static power = leakage: the power consumed by a gate whose inputs and outputs do not switch
- Sources
  - Sub-threshold leakage: at Vgs=0V, a transistor allows a small amount of current to flow
  - Gate leakage: the atomically thin gate oxide allows a small amount of current to flow
  - Junction leakage: every reverse-biased PN junctions allows a small amount of current to flow
- Typically:
  - Sub-threshold leakage > gate leakage > junction leakage

Logic-Level Low-Power Techniques

- We will introduce:
  - Pre-computation
  - Bus-encoding
  - Clock gating
  - Asynchronous sequencing (not here but in the other lecture)
  - Parallelism and pipelining
  - Memory banking

Pre-Computation

- Concept
  - Using a subset of inputs, precomputing the output logic values of the circuit one cycle before they are required.
  - If the output can be precomputed, the original logic circuits can be turned off in the succeeding clock cycle
  - The savings is determined by the cost of the added precomputation logic

Alidina et al., TVLSI'94

Pre-Computation based Power Reduction

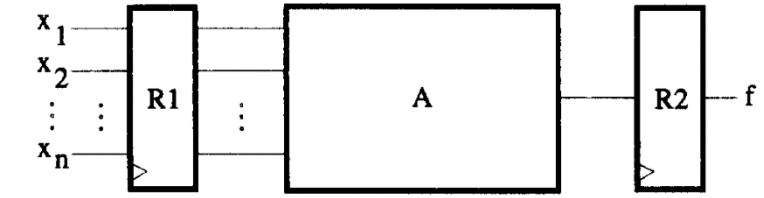
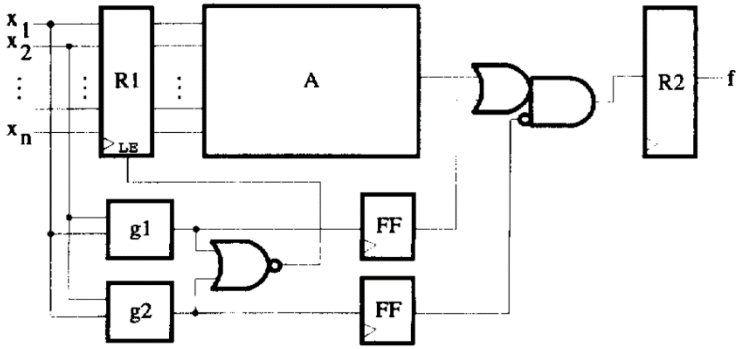


Fig. 1. Original Circuit.



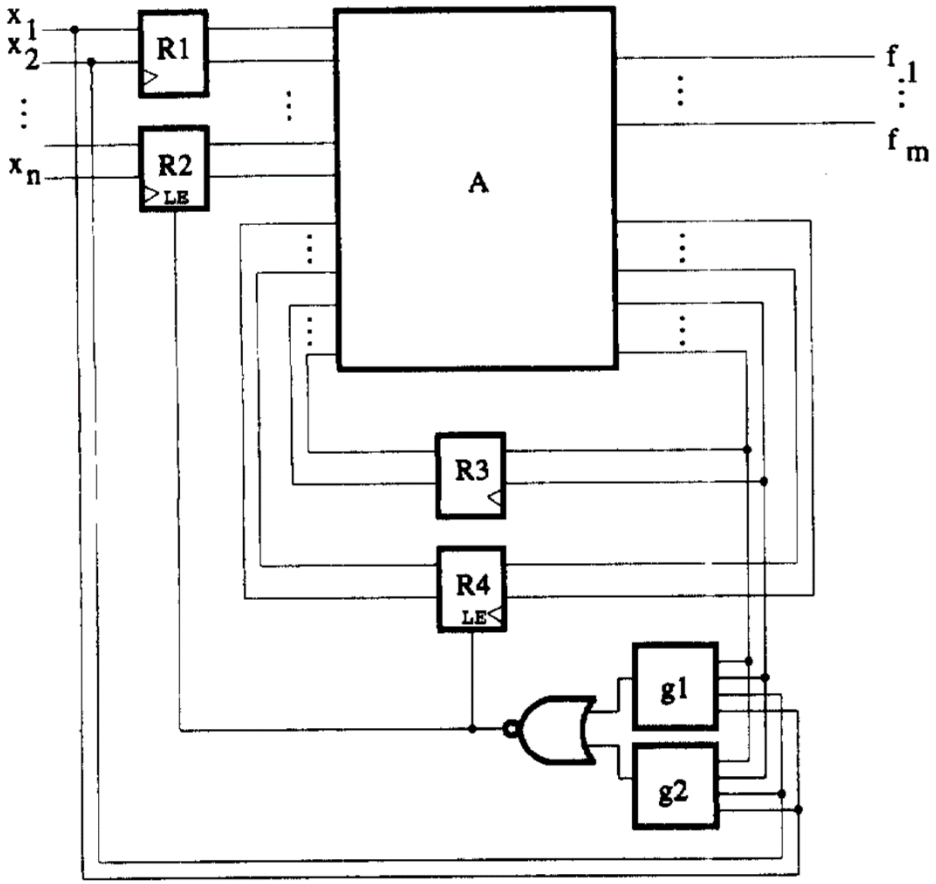
$$g_1 = 1 \Rightarrow f = 1$$

$$g_2 = 1 \Rightarrow f = 0$$

- g_1 and g_2 take only a subset of inputs
- Conditionally disable R1 and thus limit the switching of circuit A.
- Overhead: g_1 and g_2 size and the additional delay of g_1 and g_2

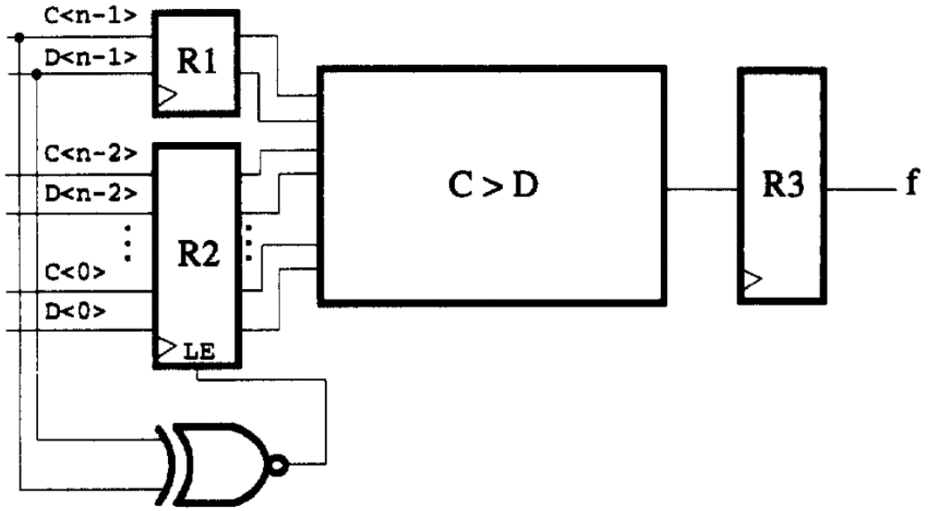
Alidina et al., TVLSI'94

Can be Applied to a FSM



Alidina et al., TVLSI'94

Example: n-bit Comparator



$$g_1 = C\langle n-1\rangle \cdot \overline{D\langle n-1\rangle}$$

$$g_2 = \overline{C\langle n-1\rangle} \cdot D\langle n-1\rangle$$

$$\overline{g_1 + g_2} = C\langle n-1\rangle \otimes D\langle n-1\rangle$$

- If $C_{n-1} \neq D_{n-1}$, whether C>D can be determined; At this condition, we don't need to consider inputs in R2
  - About 50% of the inputs are like this, resulting 50% power savings
- We can look at two bits each from C and D
  - 75% power savings

Alidina et al., TVLSI'94

Communication vs. Computation

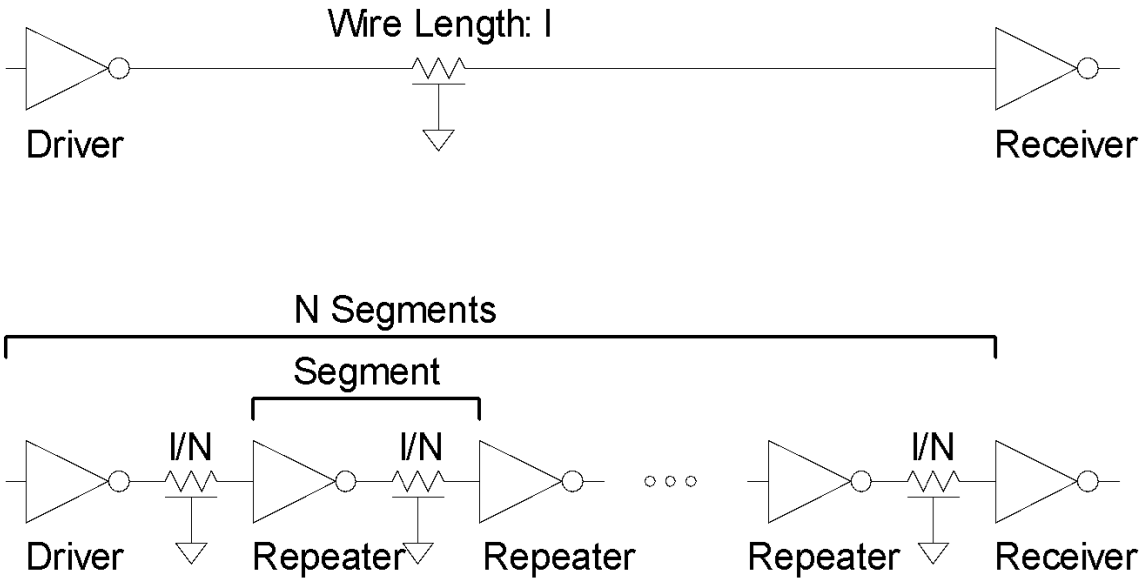
Operation	Power (0.13um)	(0.05um)
32b ALU Operation	5pJ	0.3pJ
32b Register Read	10pJ	0.6pJ
Read 32b from 8KB RAM	50pJ	3pJ
Transfer 32b across chip (10mm)	100pJ	17pJ
Execute a uP instruction (SB-1)	1.1nJ	130pJ
Transfer 32b off chip (2.5G CML)	1.3nJ	400pJ
Transfer 32b off chip (200M HSTL)	1.9nJ	1.9nJ

off-chip to global to local communication to computation

260:20:1 in 2002 → 1333:57:1 in 2010

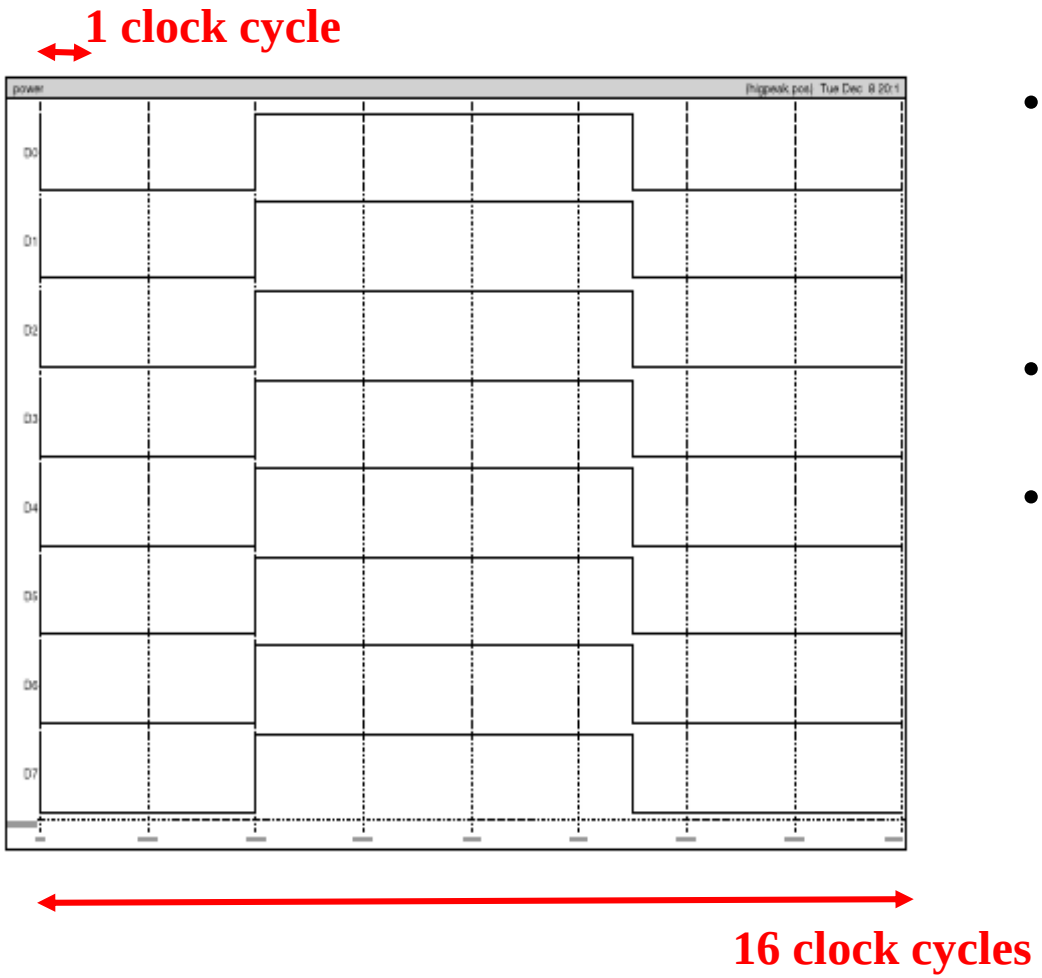
From: Dally HPCA02

Repeater Review



- Repeaters are large and there are many of them
  - Net result: lots of power consumed
- But they are good for speed and signal integrity

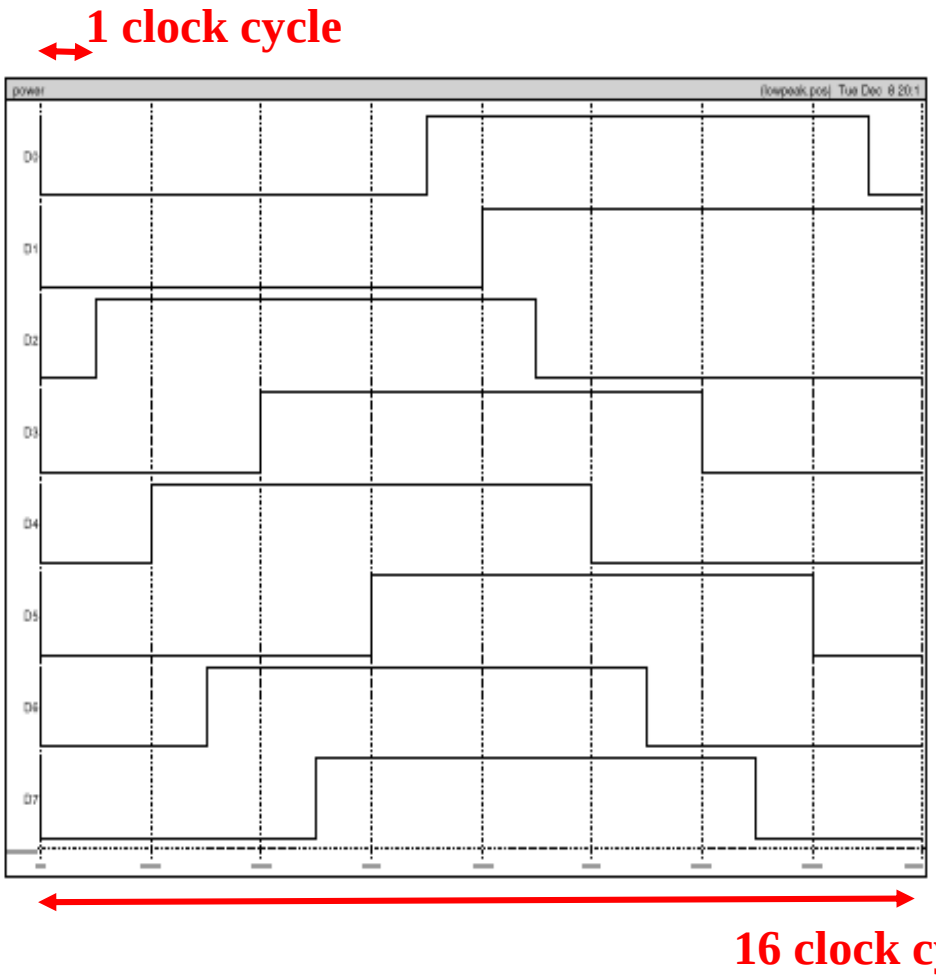
Bus-Invert Coding



- An eight-bit bus on which all eight lines toggle at the same time
- High peak power dissipation
- There are 16 transitions over 16 clock cycles (average 1 transition per clock cycle).

Stan et al., TVLSI'95

Peak Power Dissipation



An eight-bit bus on which the eight lines toggle at different moments and which has a low peak power dissipation. There are the same 16 transitions over 16 clock cycles and thus the same average power dissipation

Stan et al., TVLSI'95

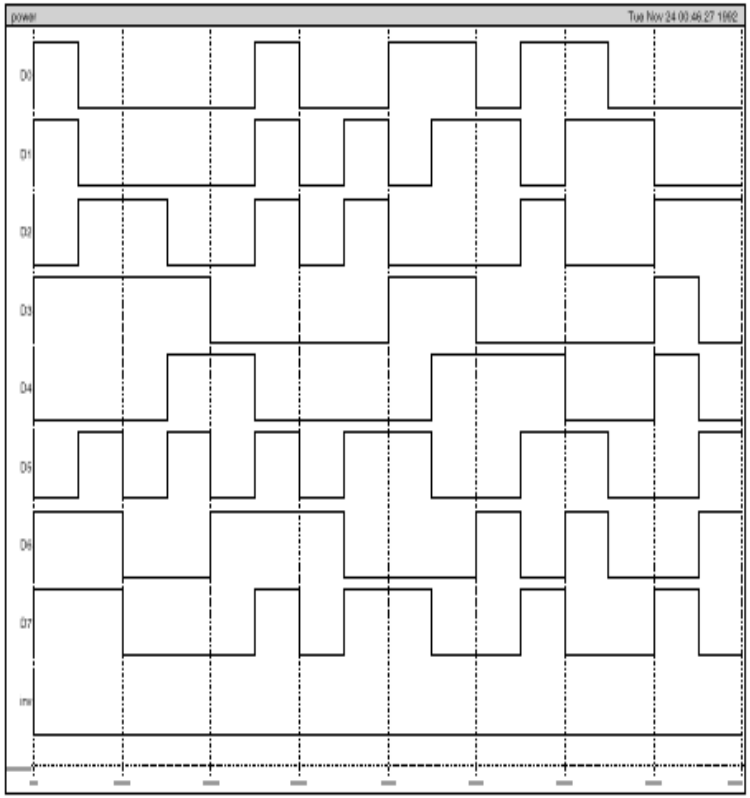
Bus-Invert - Coding for low power

The Bus-Invert method proposed here uses one extra control bit called invert. By convention then invert = 0 the bus value will equal the data value. When invert = 1 the bus value will be the inverted data value. The peak power dissipation can then be decreased by half by coding the I/O as follow

1. Compute the Hamming distance (the number of bits in which they differ) between the present bus value (also counting the present invert line) and the next data value.
2. If the Hamming distance is larger than n/2, set invert = 1 (and thus make the next bus value equal to the inverted next data value).
3. Otherwise, let invert = 0 (and let the next bus value equal to the next data value).
4. At the receiver side the contents of the bus must be conditionally inverted according to the invert line, unless the data is not stored encoded as it is (e.g. in a RAM). In any case the value of invert must be transmitted over the bus (the method increases the number of bus lines from n to n + 1).

Stan et al., TVLSI'95

Example

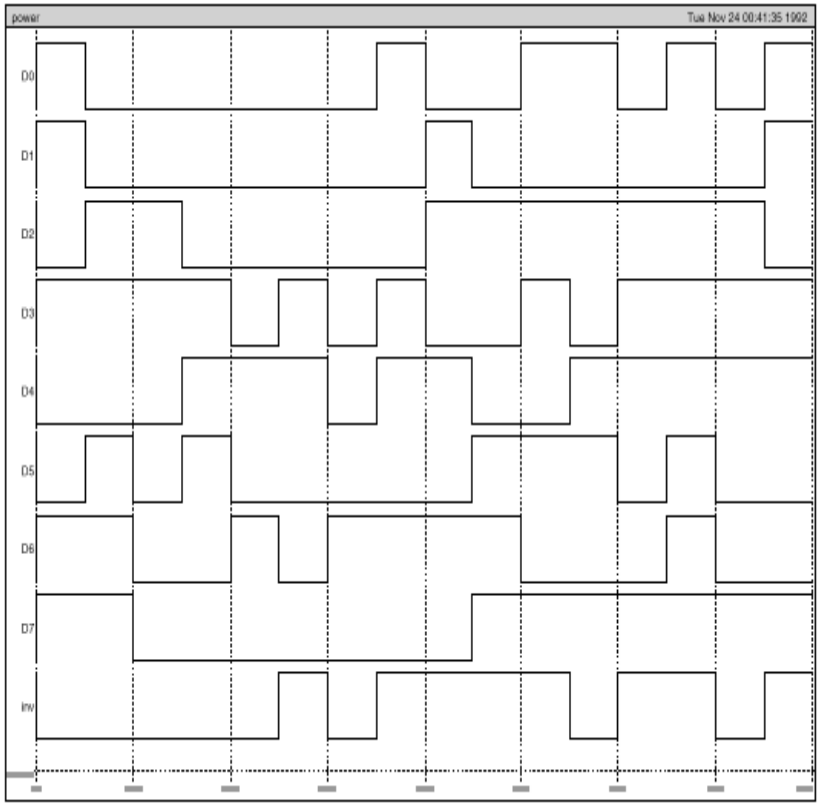


A typical eight-bit synchronous data bus. The transitions between two consecutive time-slots are clean (no glitches).

There are 64 transitions for a period of 16 time slots. This represents an average of 4 transitions per time slot, or 0.5 transitions per bus line per time slot.

Stan et al., TVLSI'95

Example



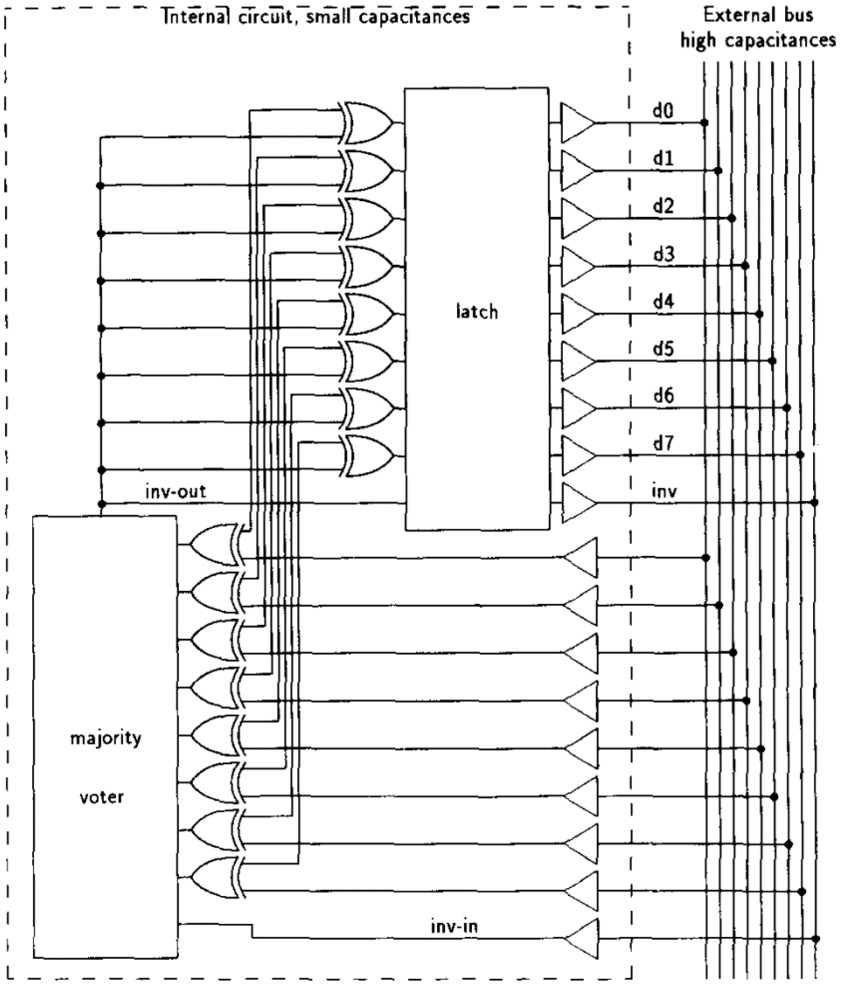
The same sequence of data coded using the Bus Invert method.

There are now only 53 transitions over a period of 16 time slots. This represents an average of 3.3 transitions per time slot, or 0.41 transitions per bus line per time slot.

The maximum number of transitions for any time slot is now 4.

Stan et al., TVLSI'95

Bus-Invert: Logic-Level Design



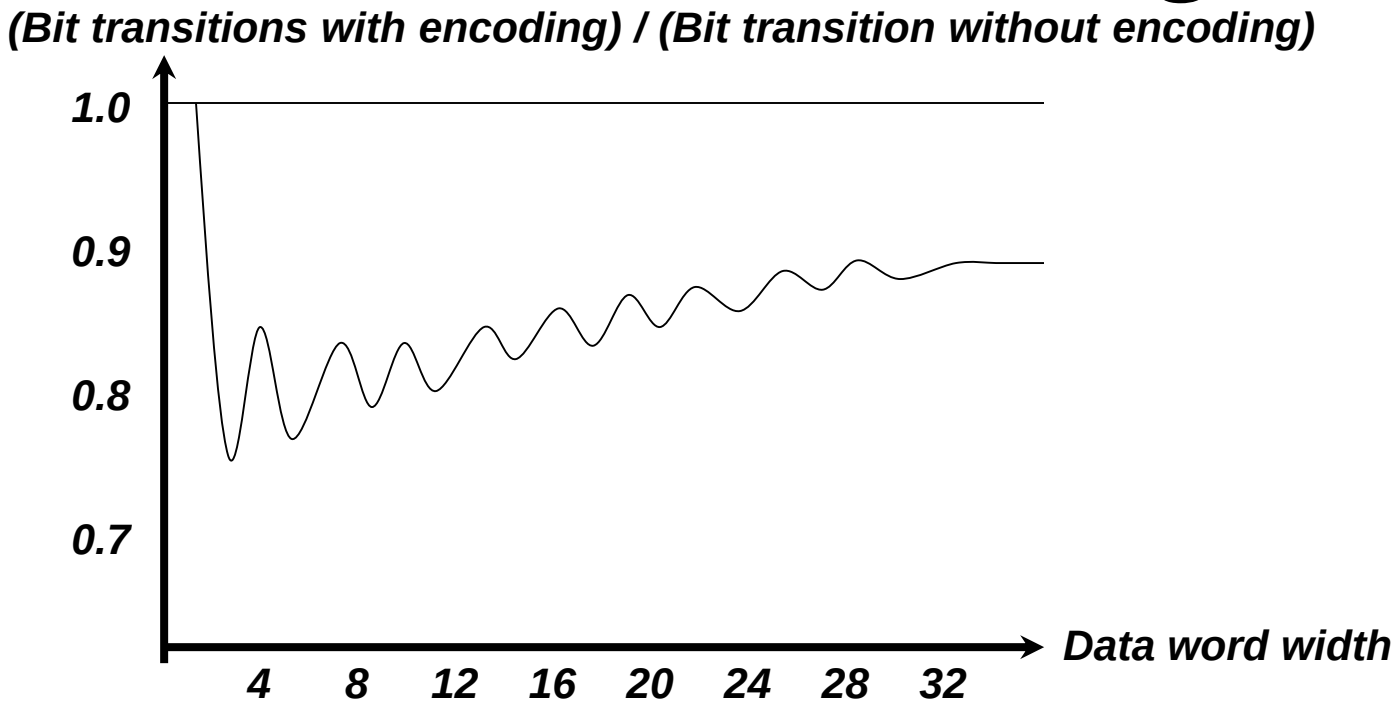
Stan et al., TVLSI'95

Bus-Inversion Coding

- Redundant coding with m = n+1
- If data word S is to be transmitted, either S or S' which is bit-wise inversion of S can be transmitted.
- Extra wire P is used to indicate the polarity.
- Decoder is memoryless, and encoder only used the current state of the wires.

Stan et al., TVLSI'95

Bus-Inversion Coding



- For large values of n, the efficacy of coding technique disappears as the ratio converges to 1.
- Dividing large bit groups to smaller groups is better.

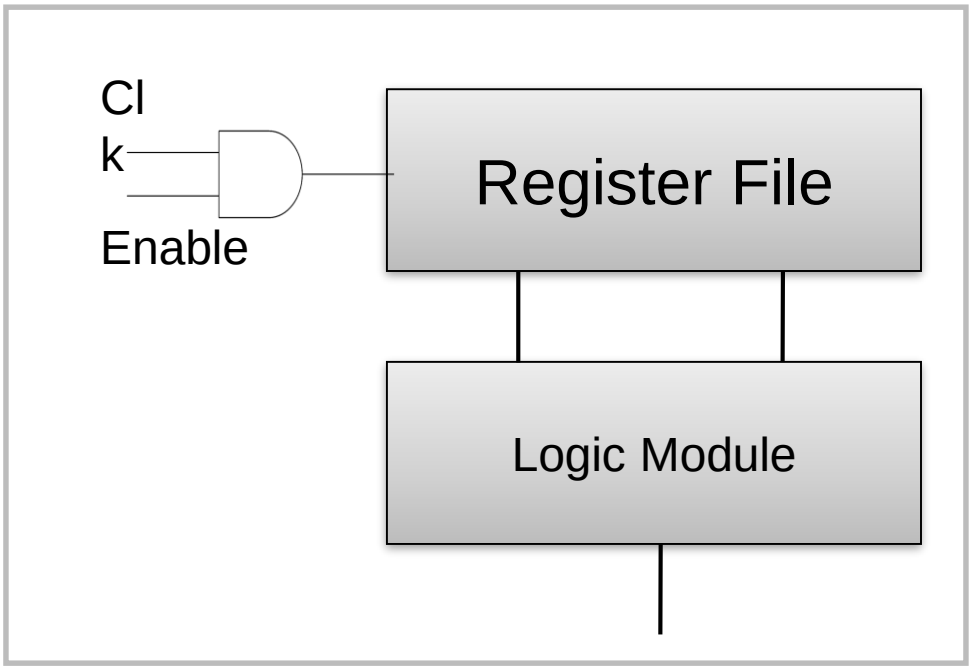
Stan et al., TVLSI'95

Clock Gating

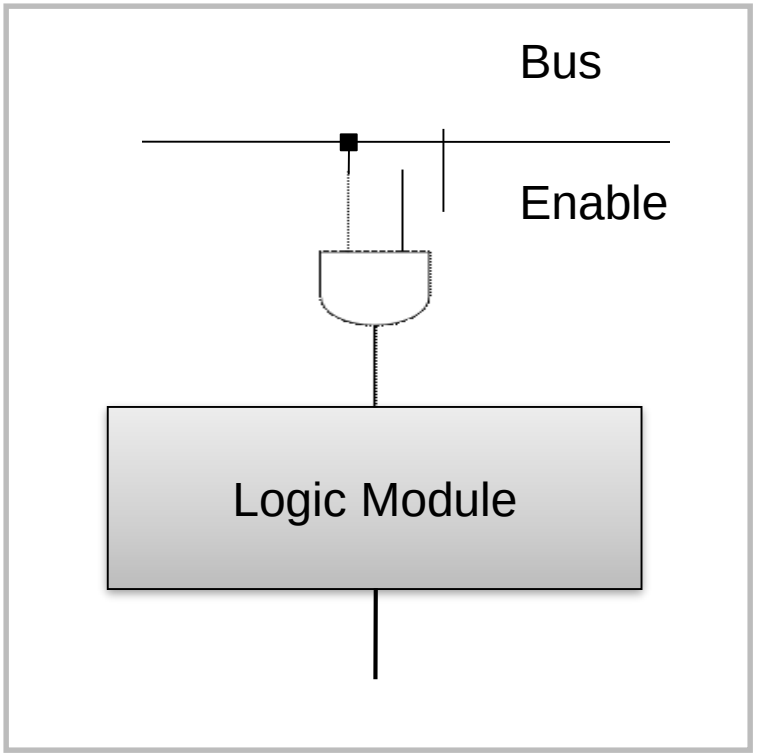
- Turn off clocks to idle modules
  - Ensure that spurious activity is set to zero
- Must ensure that data inputs to module are in stable mode
  - Primary inputs are from gated latches or registers
  - Or, disconnected from interconnect network
- Can be done at different levels of system hierarchy

Clock Gating

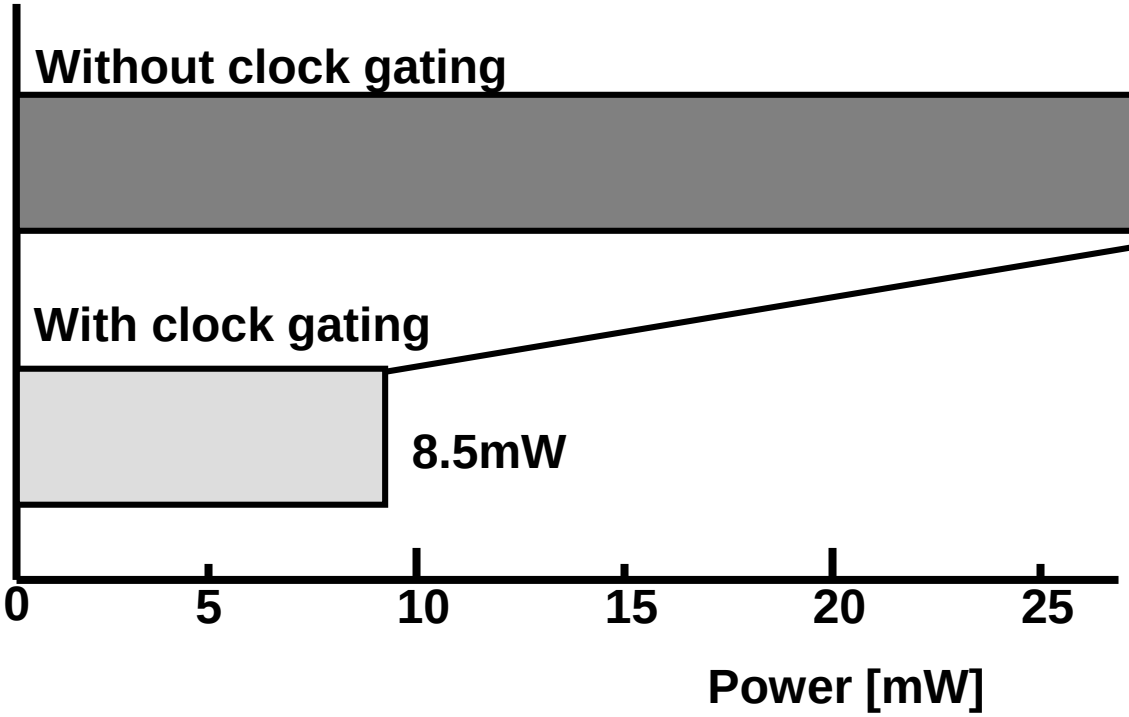
Turning off the clock for non-active components



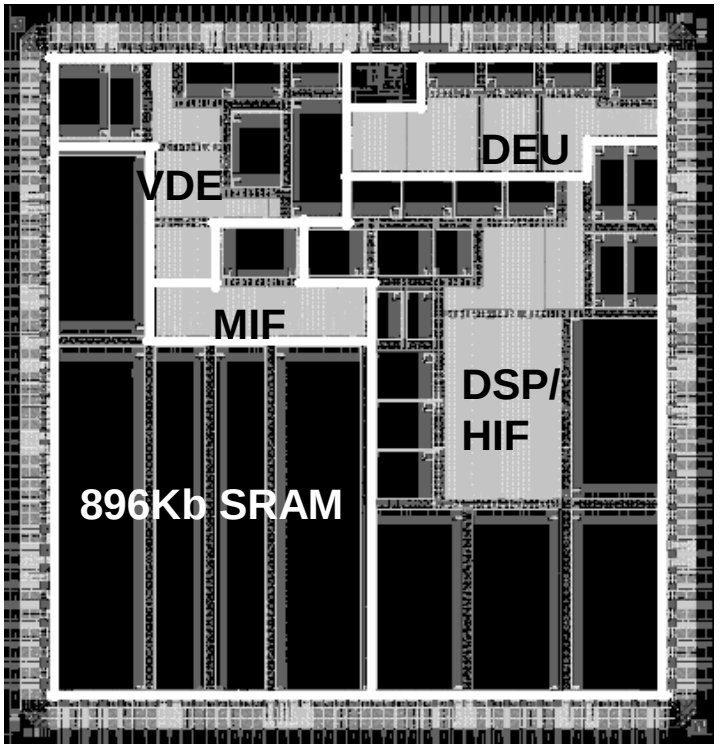
Disconnecting the inputs



Clock-gating Efficiently Reduces Power



MPEG4 decoder



90% of F/F's clock-gated.

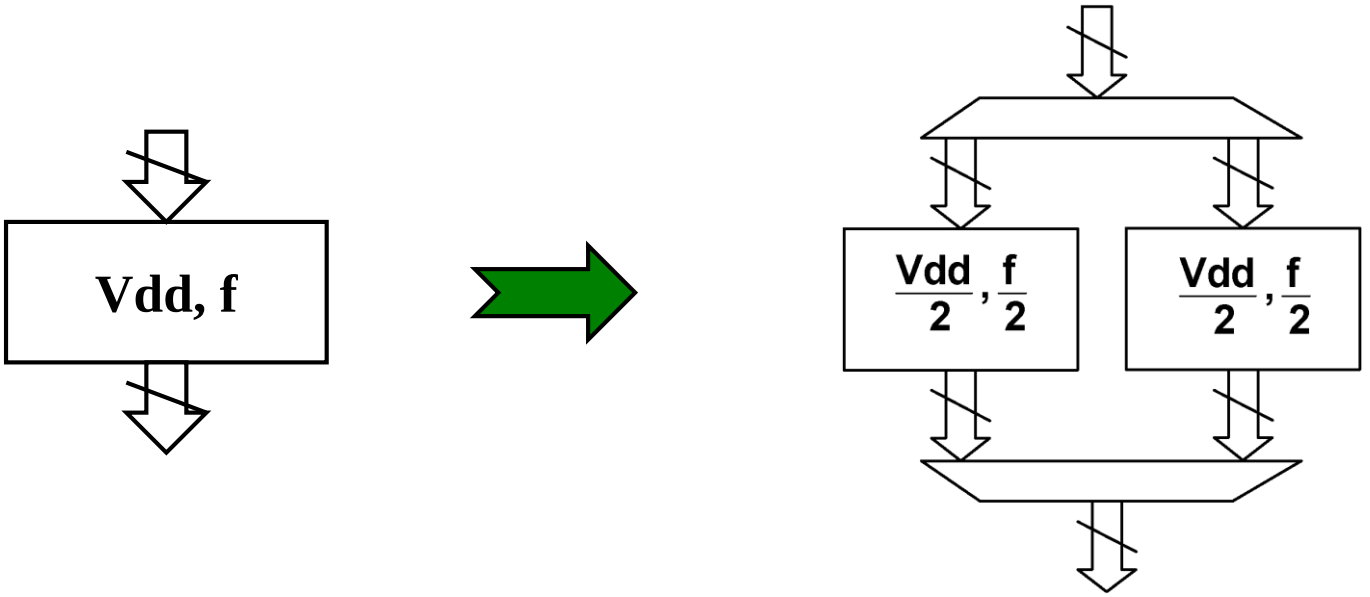
70% power reduction by clock-gating alone.

[Ref: M. Ohashi, ISSCC'02]

CAD Support for Clock Gating

- Logic synthesis tools support automatic clock gating
- After adding the AND gates, the clock distribution paths become un-balanced, thereby worsening skew
- If the AND gates are toward the source, you lose clock-gating granularity while mitigating the skew problem

Parallelism

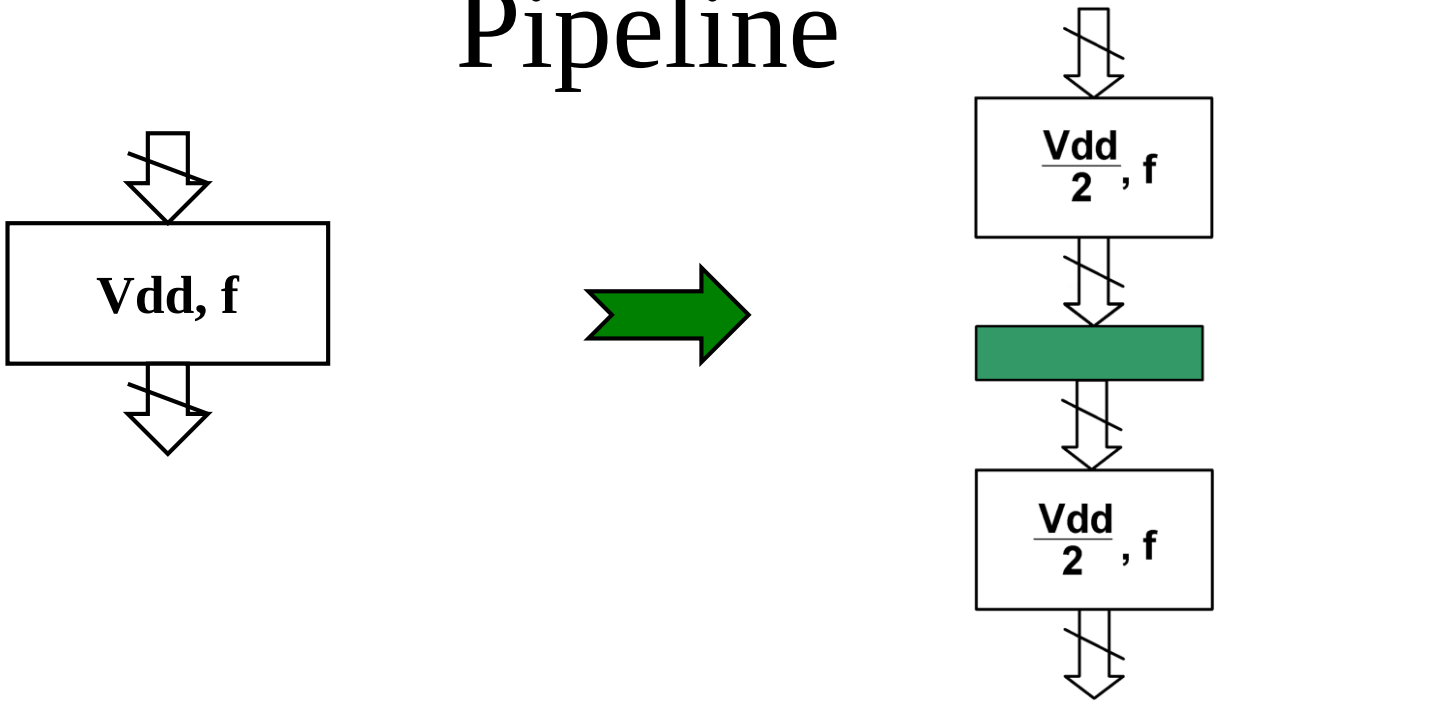


$$P_{ref} = CV_{dd}^{\ 2} f$$

$$P_{par} = 2C \cdot \left(\frac{V_{dd}}{2}\right)^2 \frac{f}{2} = \frac{1}{2^2} CV_{dd}^{\ 2} f = \frac{P_{ref}}{4}$$

- Power reduced by 2^2, not including overhead, but double the area and more leakage

Pipeline



$$P_{ref} = CV_{dd}^{\ 2} f$$

$$P_{pipe} = 2\left[\frac{C}{2}\left(\frac{V_{dd}}{2}\right)^2 f\right] = \frac{P_{ref}}{2^2}$$

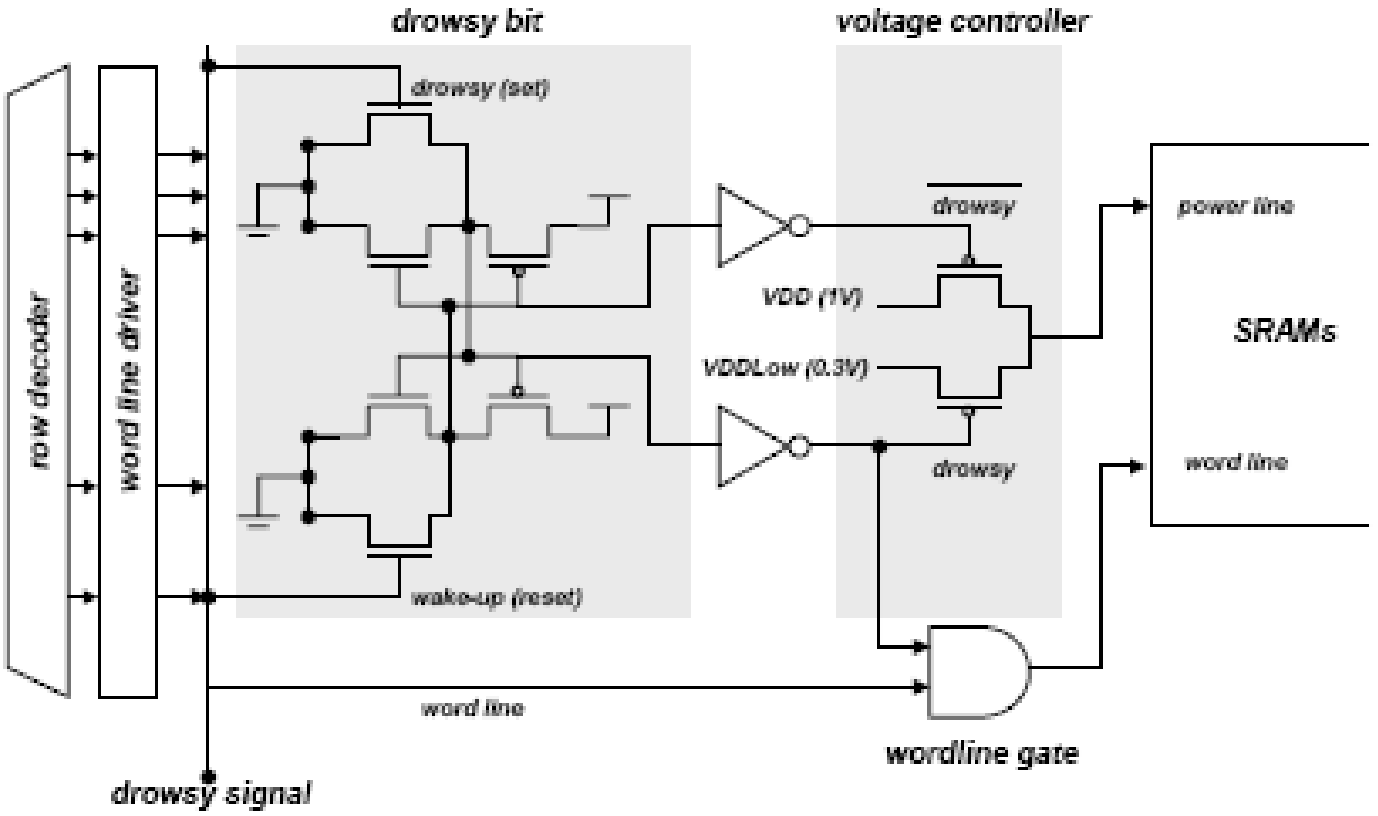
- Power reduced by 2^2, not including overhead, but lower area than parallel-arch., yet more latency
- Lower area □ less leakage

Memory Banking

- Memory power is dominated by ***leakage***
- Memory banking is mainly invented to access multiple words at the same cycle
- The banks that are not accessed can be put into low leakage state
  - MTCMOS
  - Drowsy cache [Kim, et al., Micro]
- How to put a bank into a low leakage state requires sophisticated circuit techniques

Drowsy Cache

- Reducing VDD reduces leakage
- Drowsy signal provided by architecture puts subblock into drowsy mode
- When wordline is activated, VDD is first increased (incurs delay)
- Reduces leakage by > 80%



Flautner et al., ISCA'02