

Introduction to Digital Signal Processing Systems

Shao-Yi Chien



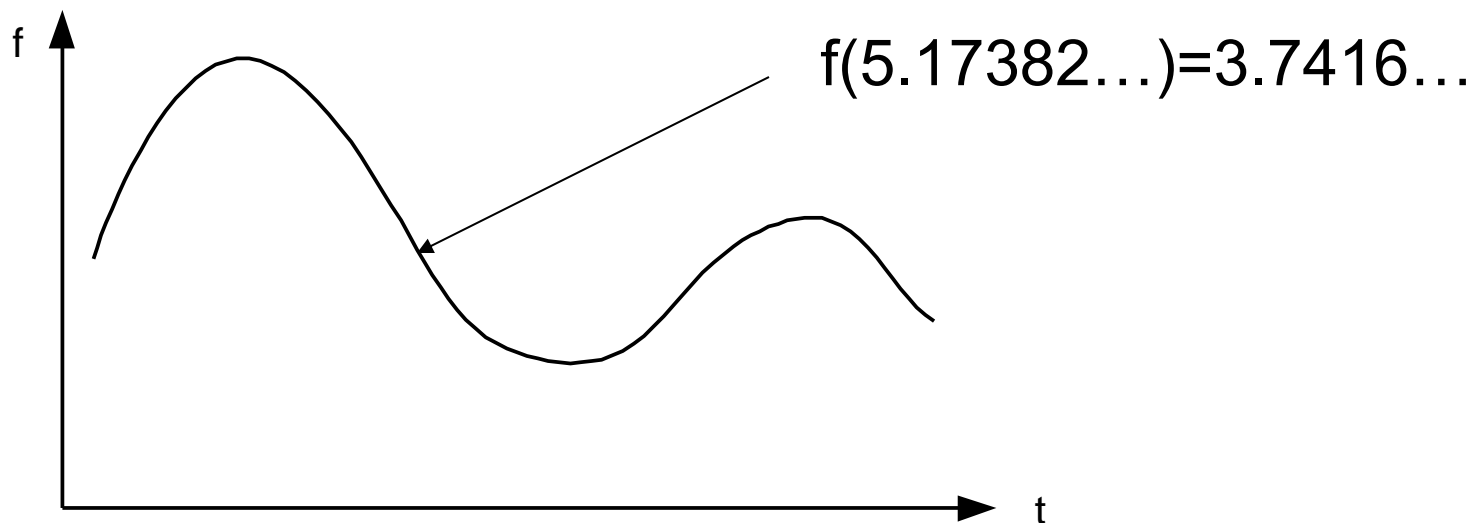
Outline

- Introduction
- Typical DSP algorithms
- Scaled CMOS technologies
- Representations of DSP algorithms



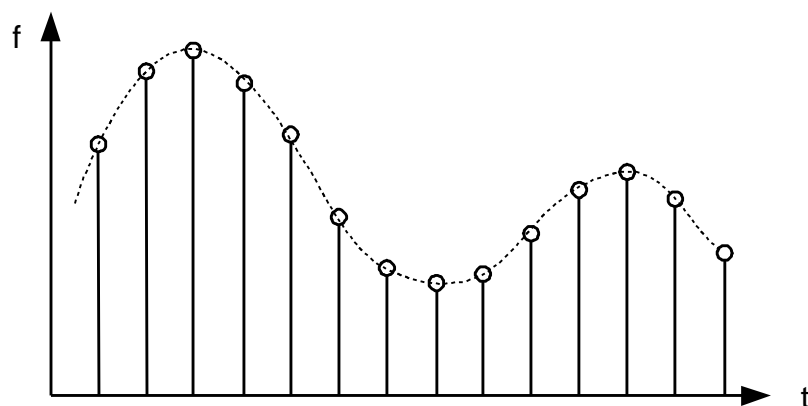
Analog Signal

- Real-world signal
- Infinite accuracy on time and magnitude

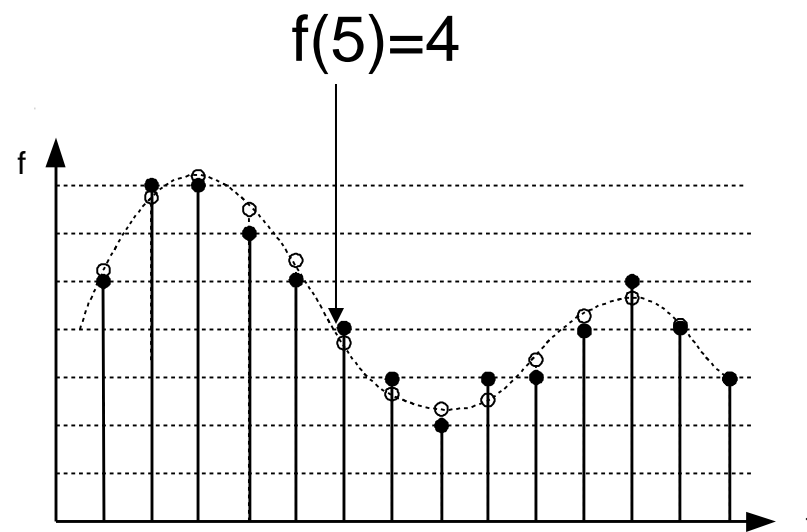


Digital Signal

- Get after sampling and quantization
- Finite accuracy on time and magnitude
- Easy to process with digital processing element



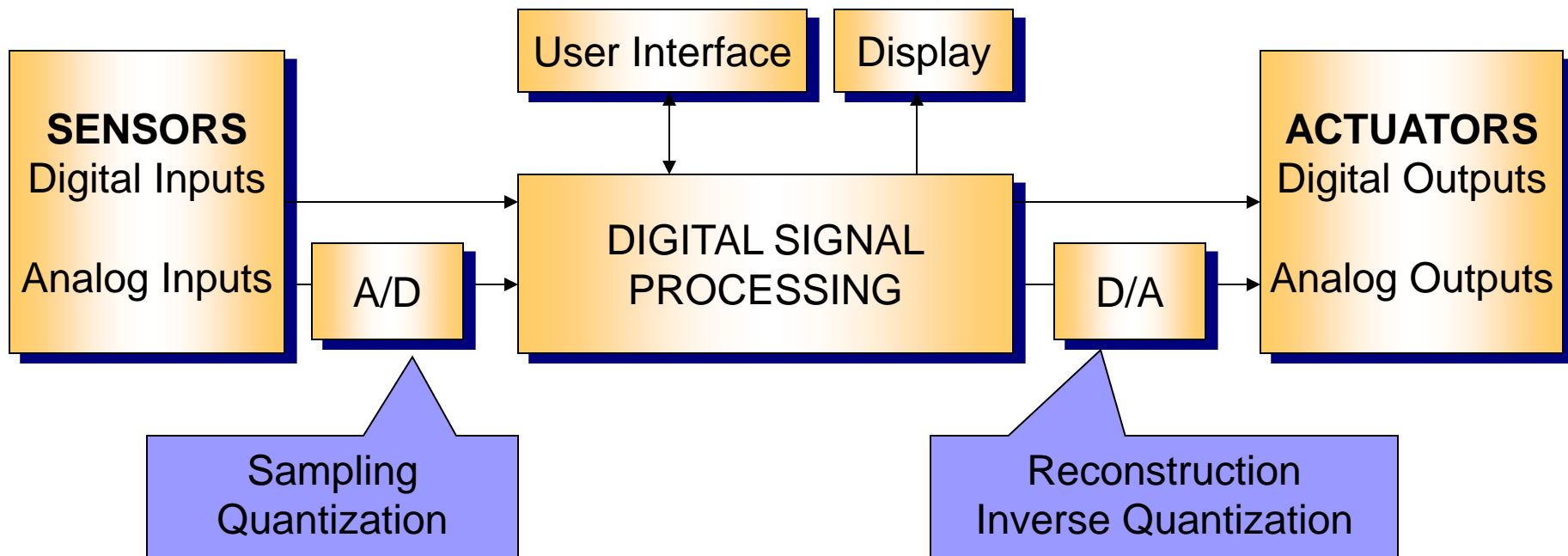
Sampling



Quantization



Typical DSP Systems





Advantages of Analog Signal Processing

- Can operate in very high frequency
- Sometimes low area
- Low power



Advantages of Digital Signal Processing (DSP)

- More robust
 - Insensitive to environment and component tolerance
- The accuracy can be controlled better
- Can cancel the noise and interference while amplifying the signal
- Predictable, repeatable behavior
 - Can be stored and recovered, transmitted and received, processed and manipulated without error



Features of DSP Systems

- Real-time throughput requirement
 - So-called hard real-time systems
- Data-driven property
- Non-terminating program



Hard Real-Time Systems



Courtesy Warner Brothers Studios

“Real-time Computing” for Formula 1





Performance Metrics of DSP Systems

- Hardware circuitry and resources (area)
- Speed of execution
- Power consumption
- Finite word length performance

Characteristics of DSP Systems (1/4)

- Data format

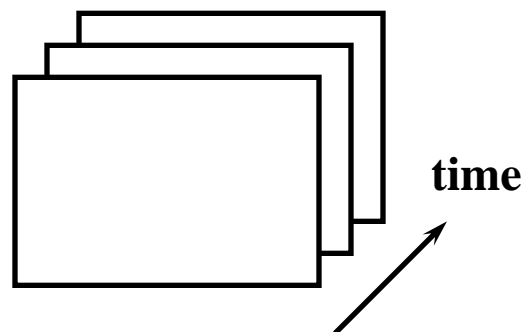
- 1D speech



- 2D image



- 3D video





Characteristics of DSP Systems

(2/4)

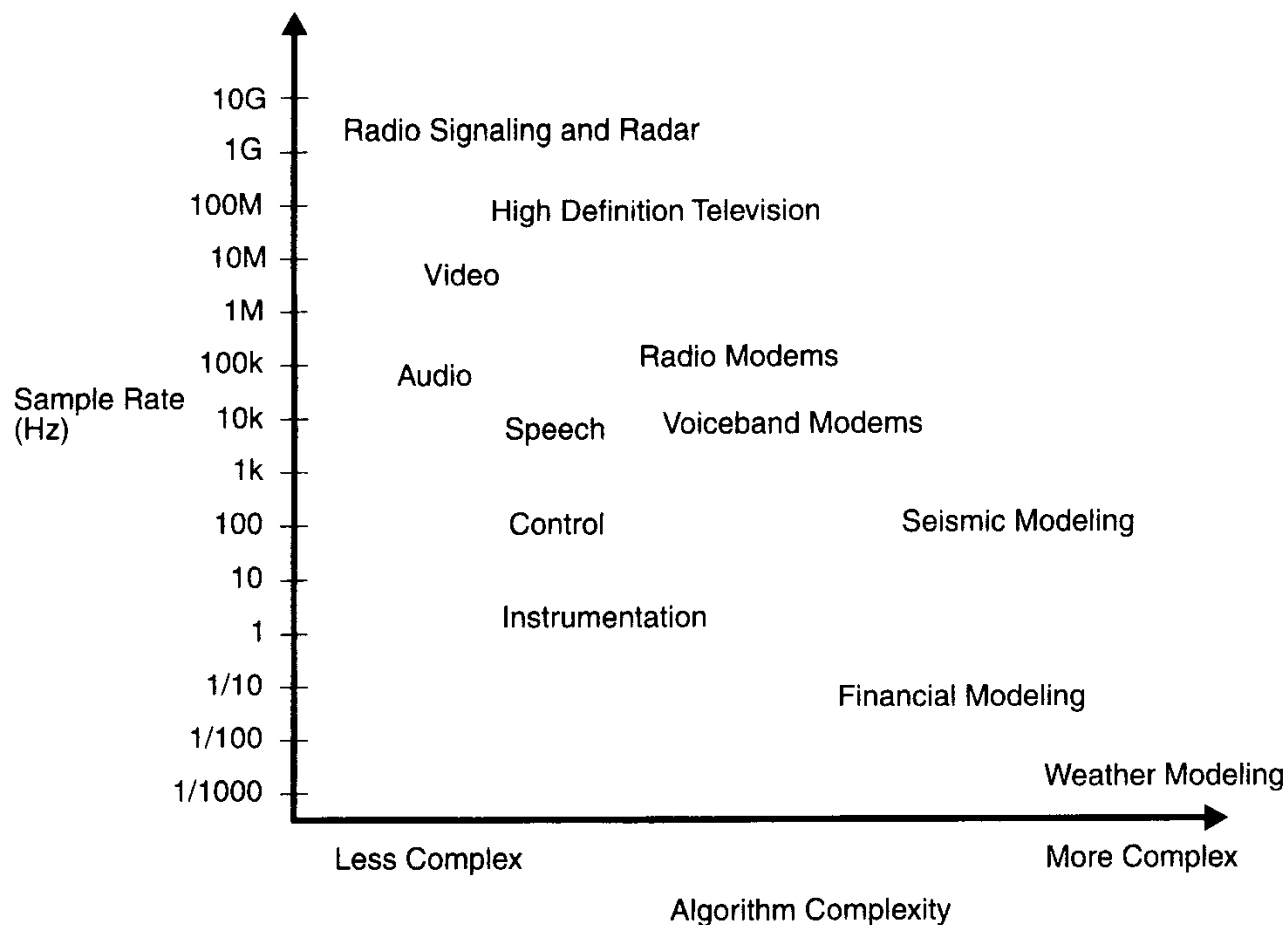
■ Algorithms

DSP Algorithm	System Application
Speech coding and decoding	Digital cellular telephones, personal communications systems, digital cordless telephones, multimedia computers, secure communications
Speech encryption and decryption	Digital cellular telephones, personal communications systems, digital cordless telephones, secure communications
Speech recognition	Advanced user interfaces, multimedia workstations, robotics, automotive applications, digital cellular telephones, personal communications systems, digital cordless telephones
Speech synthesis	Multimedia PCs, advanced user interfaces, robotics
Speaker identification	Security, multimedia workstations, advanced user interfaces
Hi-fi audio encoding and decoding	Consumer audio, consumer video, digital audio broadcast, professional audio, multimedia computers
Modem algorithms	Digital cellular telephones, personal communications systems, digital cordless telephones, digital audio broadcast, digital signaling on cable TV, multimedia computers, wireless computing, navigation, data/facsimile modems, secure communications
Noise cancellation	Professional audio, advanced vehicular audio, industrial applications
Audio equalization	Consumer audio, professional audio, advanced vehicular audio, music
Ambient acoustics emulation	Consumer audio, professional audio, advanced vehicular audio, music
Audio mixing and editing	Professional audio, music, multimedia computers
Sound synthesis	Professional audio, music, multimedia computers, advanced user interfaces
Vision	Security, multimedia computers, advanced user interfaces, instrumentation, robotics, navigation
Image compression and decompression	Digital photography, digital video, multimedia computers, video-over-voice, consumer video
Image compositing	Multimedia computers, consumer video, advanced user interfaces, navigation
Beamforming	Navigation, medial imaging, radar/sonar, signals intelligence
Echo cancellation	Speakerphones, modems, telephone switches
Spectral estimation	Signals intelligence, radar/sonar, professional audio, music

Characteristics of DSP Systems

(3/4)

■ Sample rates



Characteristics of DSP Systems

(4/4)

- Clock rates
- Numeric representations



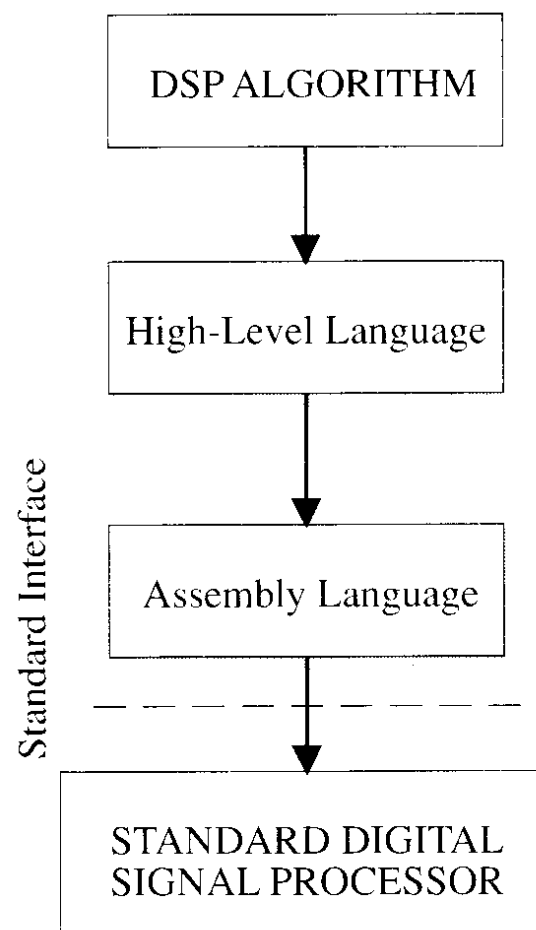
Standard Digital Signal Processors (1/2)

- Allow rapid prototyping and time-to-market
- Sometimes, the execution speed and code size is reasonably good
- Not always cost effective
- Often cannot meet the requirements of throughput, power consumption, and size

Standard Digital Signal Processors (2/2)

■ DSP Architectures

- Harvard architecture
- MAC
- Fixed-point arithmetic



Application-Specific ICs for DSP

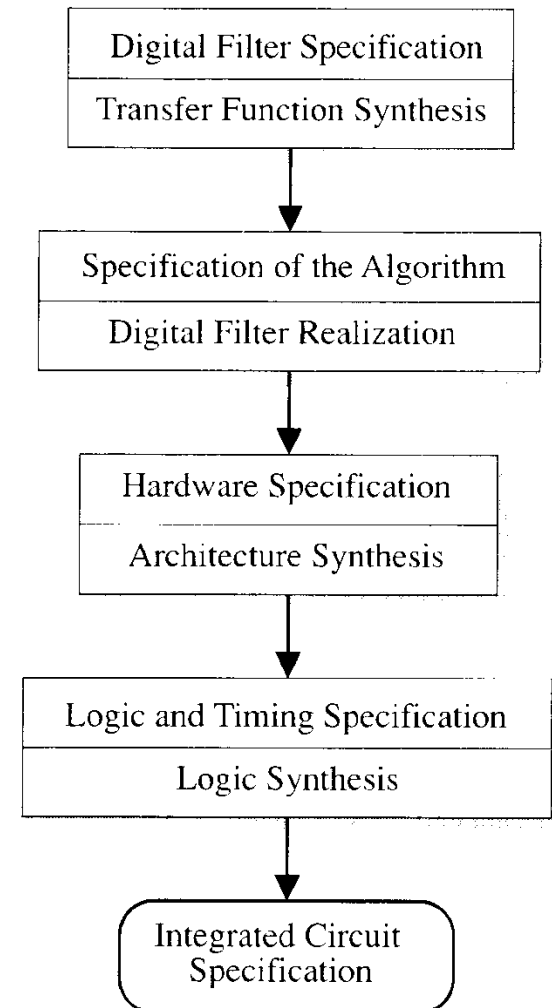
(1/2)

- Better performances
 - Processing capacity
 - Power consumption
 - Pin-restriction problem
- Main problem is the system is very complex to design
 - Long time-to-market

Application-Specific ICs for DSP

(2/2)

- Large design space
- Hard to find optimal solution
- System → specification → algorithm → hardware architecture → logic implementation → VLSI implementation





Typical DSP Algorithms

- Convolution
- Correlation
- Digital filters
- Adaptive filters
- Motion estimation
- Discrete cosine transform (DCT)
- Vector quantization (VQ)
- Viterbi algorithm and dynamic programming
- Decimator and expander
- Wavelets and filter banks



Convolution (1/2)

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n - k).$$

- Can be used to describe the behavior of a linear time-invariant systems
 - $x(n)$: input signal
 - $y(n)$: output signal
 - $h(n)$: unit-sample response



Convolution (2/2)

- Finite impulse response (FIR) system

$$h(n) = \frac{1}{M_1 + M_2 + 1} \sum_{-M_1}^{M_2} \delta(n - k)$$

- Infinite impulse response (IIR) system

$$h(n) = \sum_{k=-\infty}^n \delta(k)$$



Digital Filters

- LTI, causal filter

$$y(n) = - \sum_{k=1}^N a_k y(n - k) + \sum_{k=0}^{M-1} b_k x(n - k).$$

- M-tap finite impulse response filter

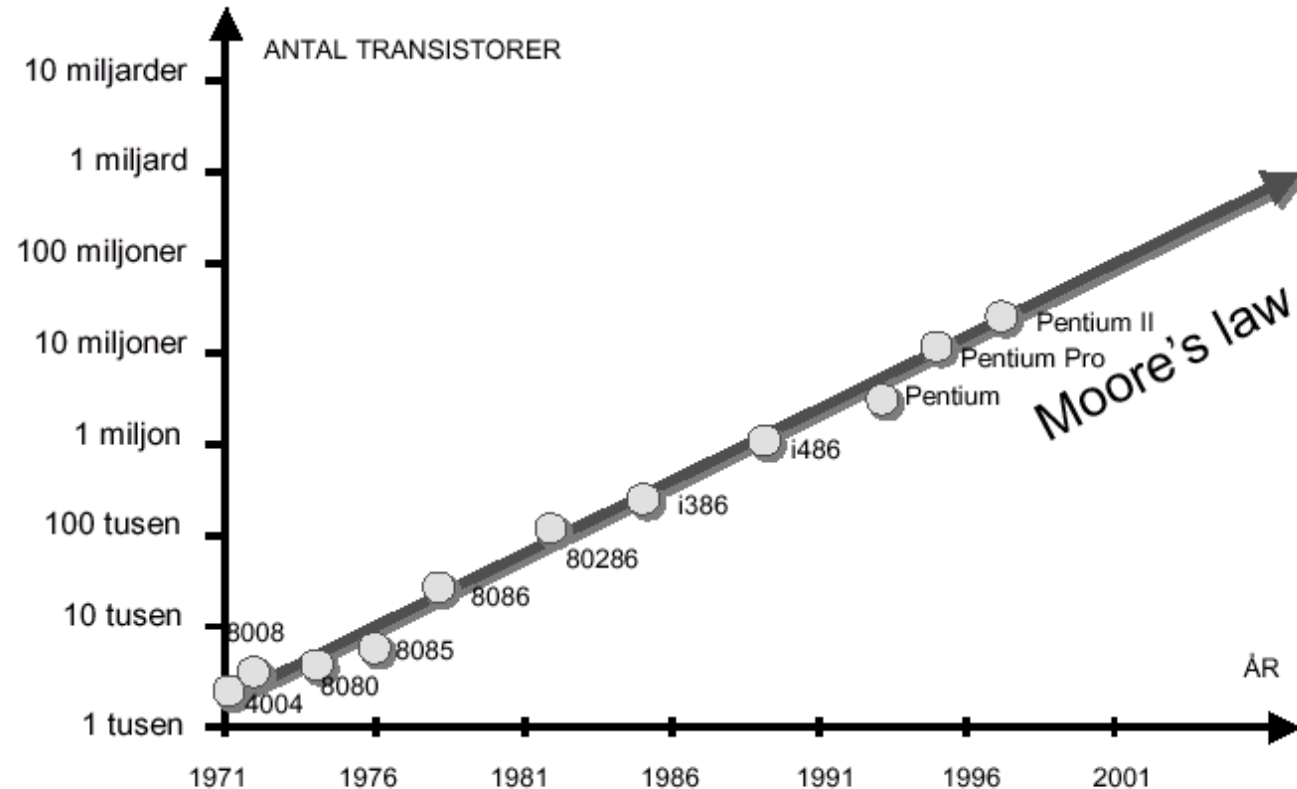
$$y(n) = \sum_{k=0}^{M-1} b_k x(n - k)$$

Chip Development

Moore's law:
The number of transistors per chip doubles every 18 months.



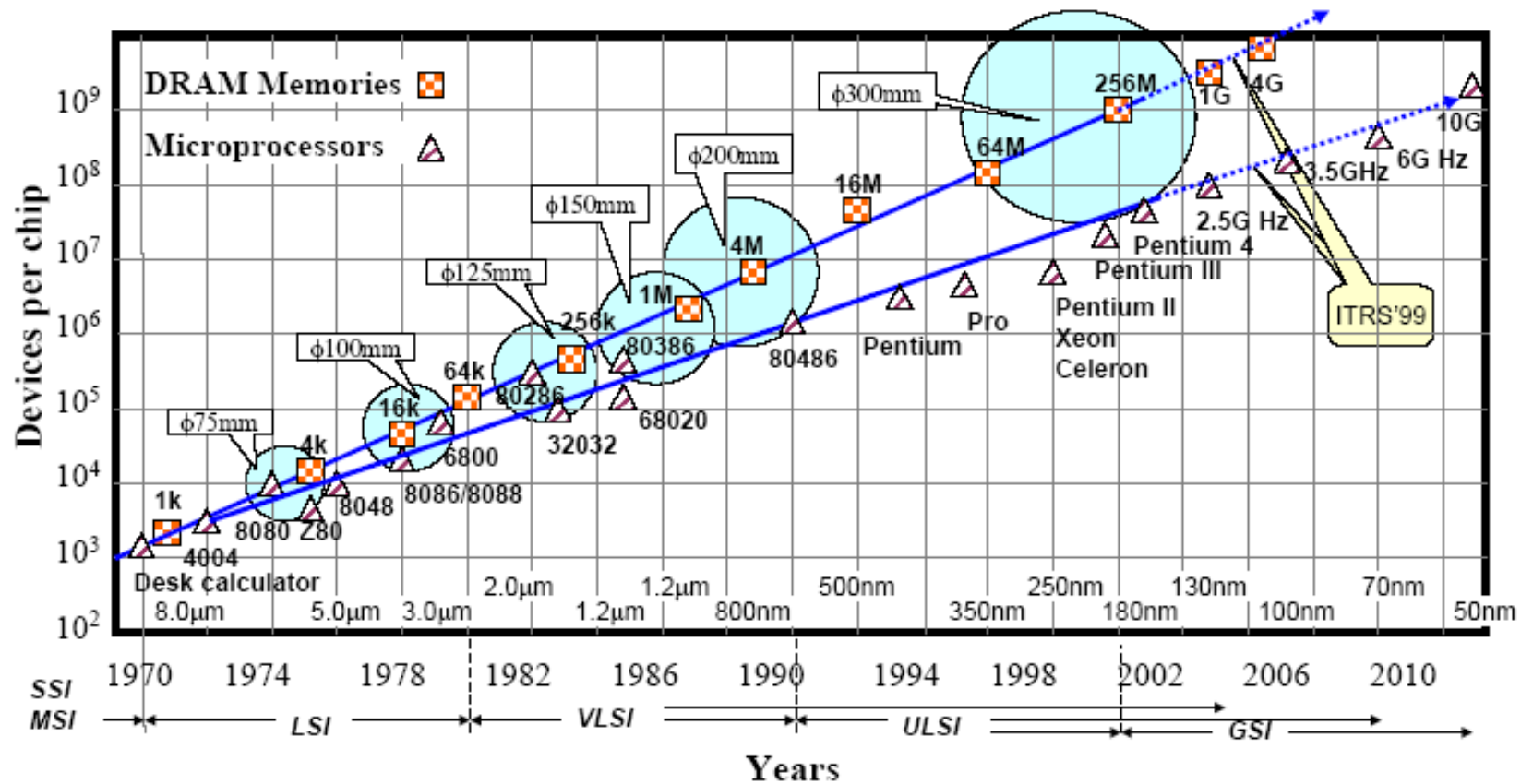
Gordon Moore
One of the founders of Intel



Technology roadmap:

<http://notes.sematech.org/ntrs/PubINTRS.nsf>

Moore's Law





Scaled CMOS technology (Moore's Law) (1/3)

Year of Production:	2001	2003	2005	2007	2010	2016
DRAM Half-Pitch [nm]:	130	100	80	65	45	22
Overlay Accuracy [nm]:	46	35	28	23	18	9
MPU Gate Length [nm]:	90	65	45	35	25	13
CD Control [nm]:	8	5.5	3.9	3.1	2.2	1.1
T _{OX} (equivalent) [nm]:	1.3-1.6	1.1-1.6	0.8-1.3	0.6-1.1	0.5-0.8	0.4-0.5
Junction Depth [nm]:	48-95	33-66	24-47	18-37	13-26	7-13
Metal Cladding [nm]:	16	12	9	7	5	2.5
Inter-Metal Dielectric K:	3.0-3.6	3.0-3.6	2.6-3.1	2.3-2.7	2.1	1.8



Scaled CMOS technology (Moore's Law) (2/3)

Year of first DRAM shipment	1995	1998	2001	2004	2007	2010
Minimum feature of size (um)	0.35	0.25	0.18	0.13	0.10	0.07
Memory in bits/chip (DRAM/FLASH)	64M	256M	1G	4G	16G	64G
Microprocessor transistor per chip (2.3 times per generation)	12M	28M	64M	150M	350M	800M
ASIC (gate per chip)	5M	14M	26M	50M	210M	430M
Chip frequency (MHz) for a high-performance on-chip clock	300	450	600	800	1,000	1,100
Maximum number of wiring levels (logic), on chip	4-5	5	5-6	6	6-7	7-8
Power supply voltage (V) for desktop	3.3	2.5	1.8	1.5	1.2	0.9
Maximum power for high performance with heat sink (W)	80	100	120	140	160	180

Source: SIA (Semiconductor Industry Association) road map
ITRS: International Technology Roadmap for Semiconductors
<http://www.itrs.net/>



Scaled CMOS technology (Moore's Law) (3/3)

Year of Production	2013	2015	2017	2019	2021	2023	2025	2028
Logic Industry "Node Name" Label	"16/14"	"10"	"7"	"5"	"3.5"	"2.5"	"1.8"	
Logic ½ Pitch (nm)	40	32	25	20	16	13	10	7
Flash ½ Pitch [2D] (nm)	18	15	13	11	9	8	8	8
DRAM ½ Pitch (nm)	28	24	20	17	14	12	10	7.7
FinFET Fin Half-pitch (new) (nm)	30	24	19	15	12	9.5	7.5	5.3
FinFET Fin Width (new) (nm)	7.6	7.2	6.8	6.4	6.1	5.7	5.4	5.0
6-t SRAM Cell Size(um ²) [@60f2]	0.096	0.061	0.038	0.024	0.015	0.010	0.0060	0.0030
MPU/ASIC HighPerf 4t NAND Gate Size(um ²)	0.248	0.157	0.099	0.062	0.039	0.025	0.018	0.009
4-input NAND Gate Density (Kgates/mm) [@155f2]	4.03E+03	6.37E+03	1.01E+04	1.61E+04	2.55E+04	4.05E+04	6.42E+04	1.28E+05
Flash Generations Label (bits per chip) (SLC/MLC)	64G /128G	128G /256G	256G / 512G	512G / 1T	512G / 1T	1T / 2T	2T / 4T	4T / 8T
Flash 3D Number of Layer targets (at relaxed Poly halfpitch)	16-32	16-32	16-32	32-64	48-96	64-128	96-192	192-384
Flash 3D Layer half-pitch targets (nm)	64nm	54nm	45nm	30nm	28nm	27nm	25nm	22nm
DRAM Generations Label (bits per chip)	4G	8G	8G	16G	32G	32G	32G	32G
450mm Production High Volume Manufacturing Begins (100Kwspm)				2018				
Vdd (High Performance, high Vdd transistors)**	0.86	0.83	0.80	0.77	0.74	0.71	0.68	0.64
1/(CVI) (1/psec) **	1.13	1.53	1.75	1.97	2.10	2.29	2.52	3.17
On-chip local clock MPU HP [at 4% CAGR]	5.50	5.95	6.44	6.96	7.53	8.14	8.8	9.9
Maximum number wiring levels [unchanged]	13	13	14	14	15	15	16	17
MPU High-Performance (HP) Printed Gate Length (GLpr) (nm) **	28	22	18	14	11	9	7	5
MPU High-Performance Physical Gate Length (GLph) (nm) **	20	17	14	12	10	8	7	5
ASIC/Low Standby Power (LP) Physical Gate Length (nm) (GLph)**	23	19	16	13	11	9	8	6

** Note: from the PIDS working group data; however, the calibration of Vdd, GLph, and I/CV is ongoing for improved targets in 2014 ITRS work



DSP and VLSI

■ Modern DSP

- Well suited to VLSI implementation
- Feasible or economically viable only if implemented using VLSI technologies

■ VLSI

- Large investment → need large volume of products
 - Communication
 - Consumer applications
- Necessary performance requirement (especially real-time requirement)
 - DSP systems are hard real-time systems



Example: the Chip for PS2

Ref: K. Kutaragi, The Future of Computing for "Real-Time Entertainment," ISSCC2006

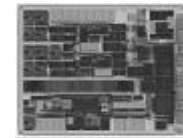
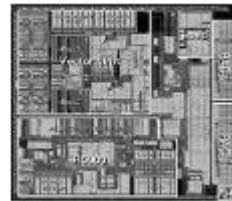
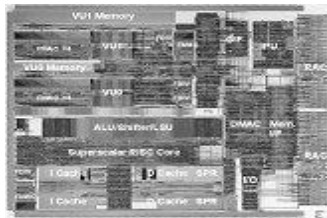
Emotion Engine

'98
239 mm²
@0.25um

'99
110 mm²
@0.18um

'01
75 mm²
@0.14um

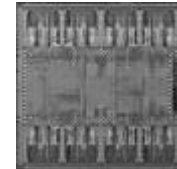
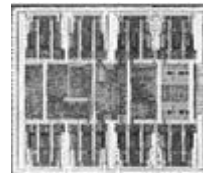
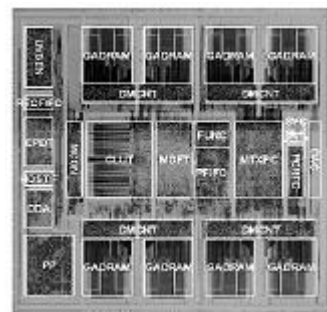
'04
87 mm²
@90nm



'98
279 mm²
@0.25um

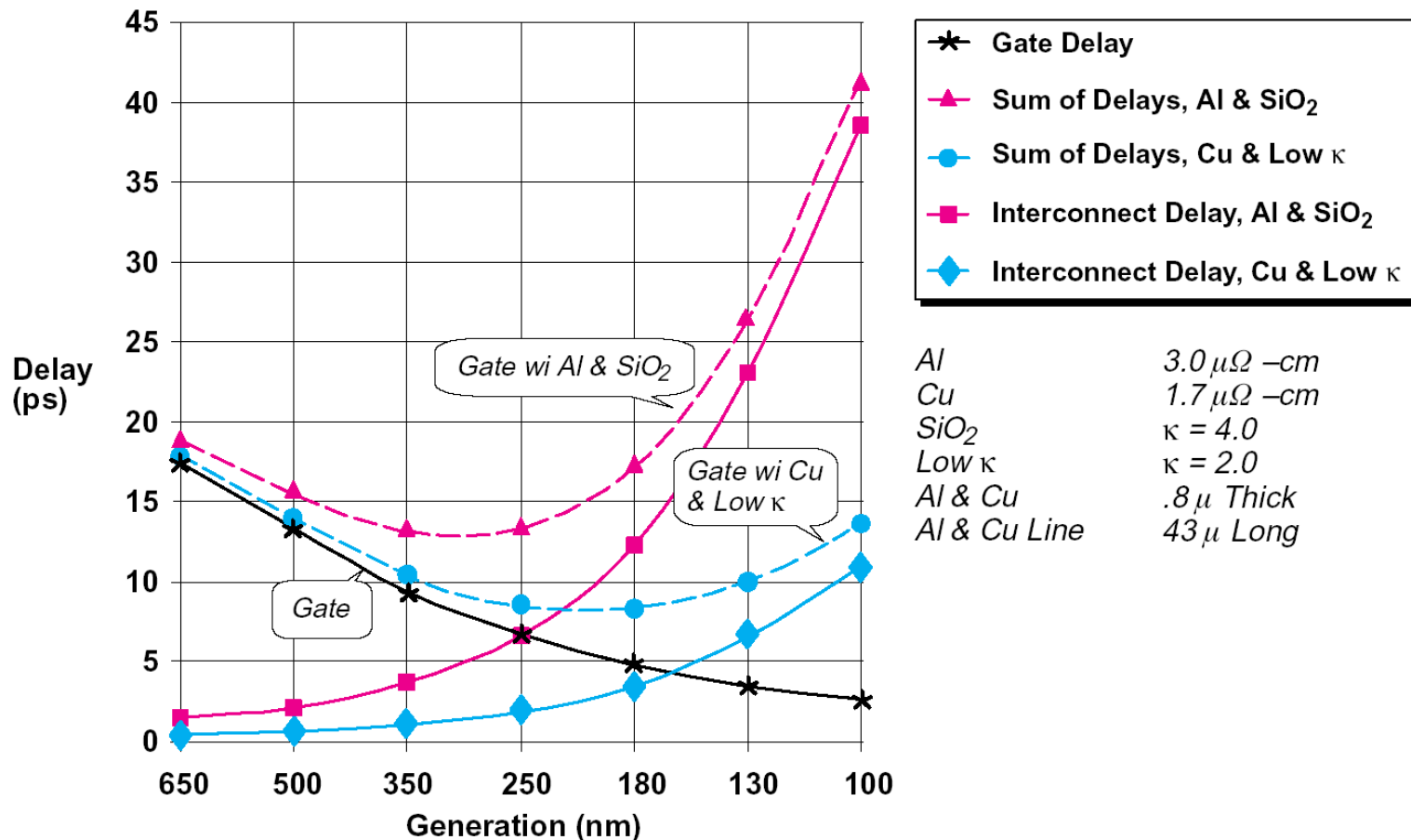
'99
108 mm²
@0.18um

'02
77 mm²
@0.16um





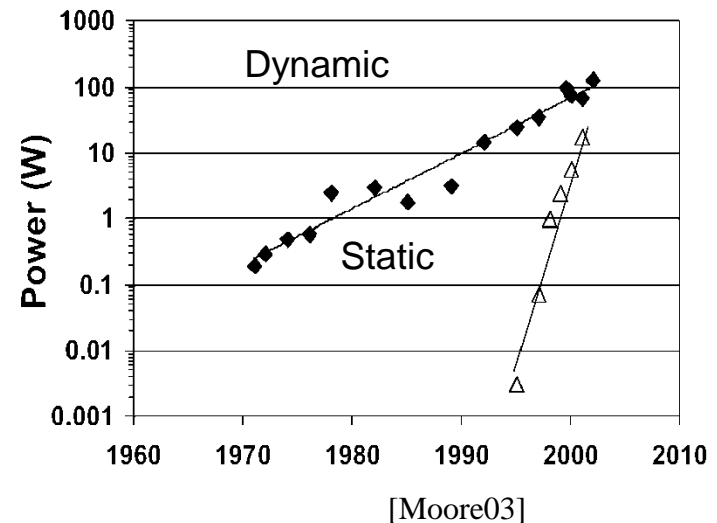
Problems: Interconnection





Problems: Increasing Static Power

- V_{DD} decreases
 - Save dynamic power
 - Protect thin gate oxides and short channels
 - No point in high value because of velocity sat.
- V_t must decrease to maintain device performance
- But this causes exponential increase in OFF leakage
- Major future challenge



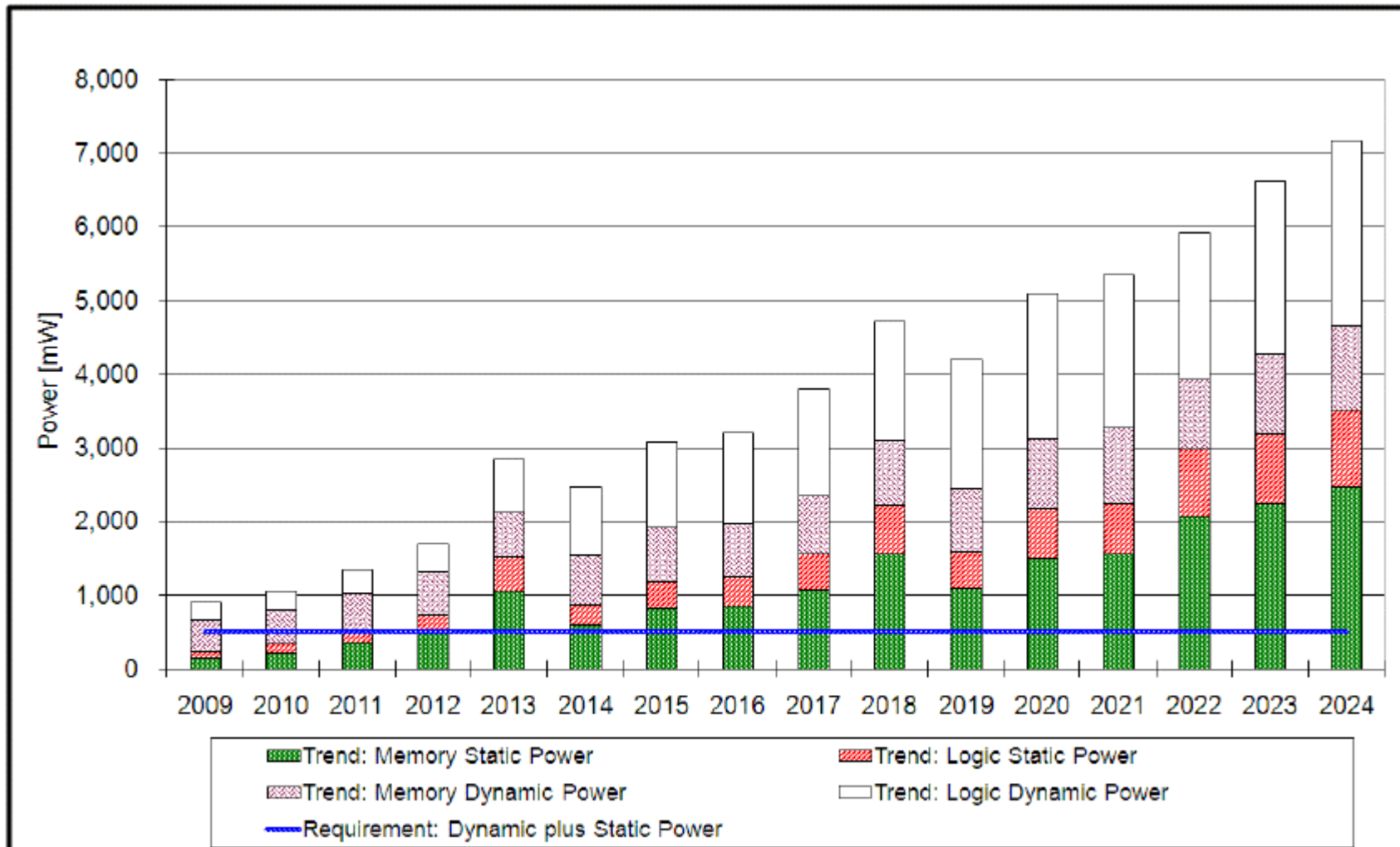
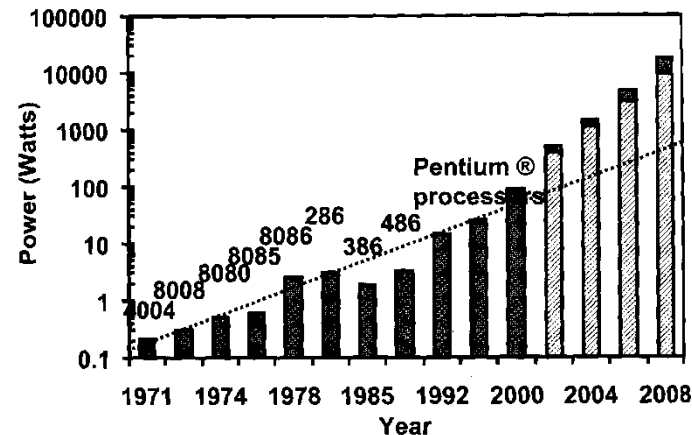


Figure SYSD6 SOC Consumer Portable Power Consumption Trends—UPDATED

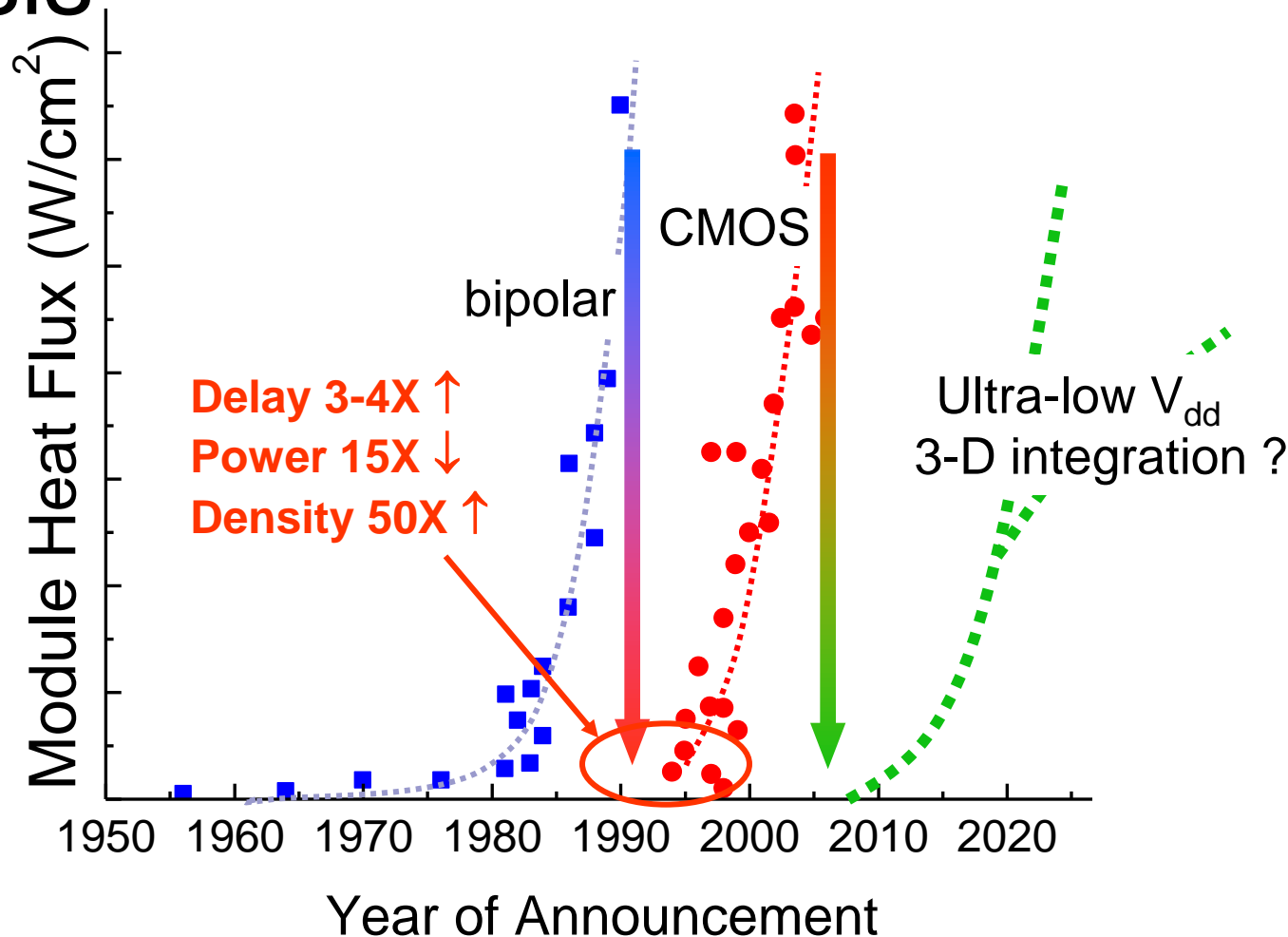
Problems: Power Density

- Intel VP Patrick Gelsinger (ISSCC 2001)
 - If scaling continues at present pace, by 2005, high speed processors would have power density of nuclear reactor, by 2010, a rocket nozzle, and by 2015, surface of sun.
 - “Business as usual will not work in the future.”
- Intel stock dropped 8% on the next day
- But attention to power is increasing



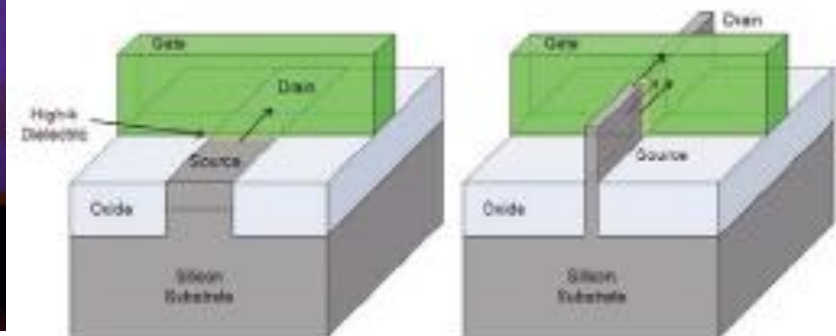


Problems: Scaling and the Power Crisis



After: R. Schmidt *et al.*, IBM J. R&D, (2002).

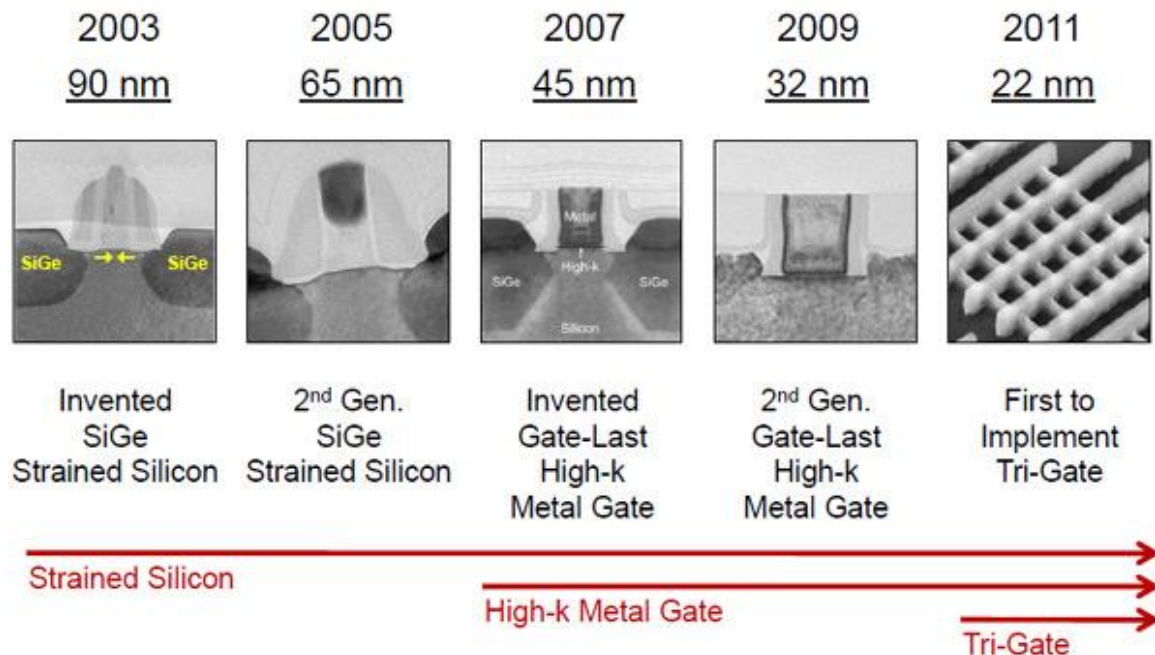
FinFET





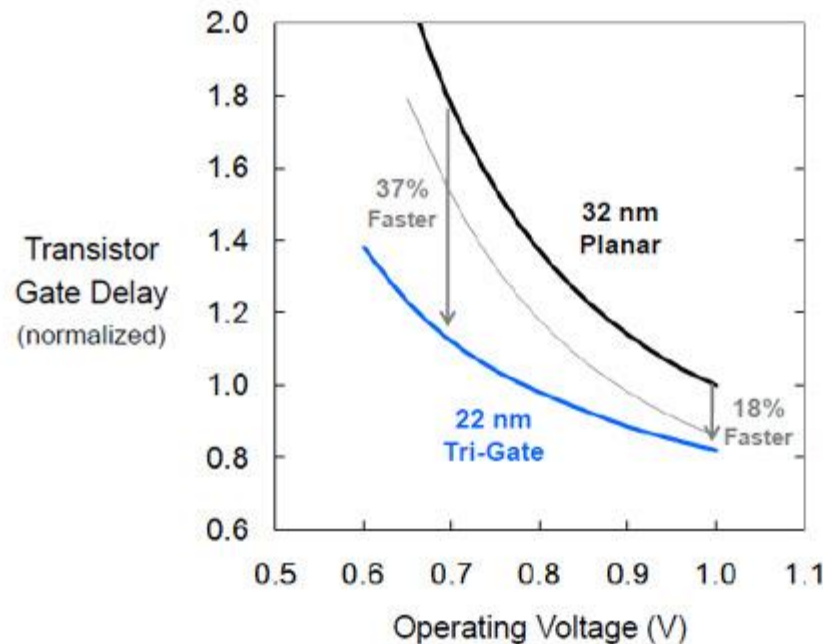
FinFET

Intel Transistor Leadership



FinFET

Transistor Gate Delay



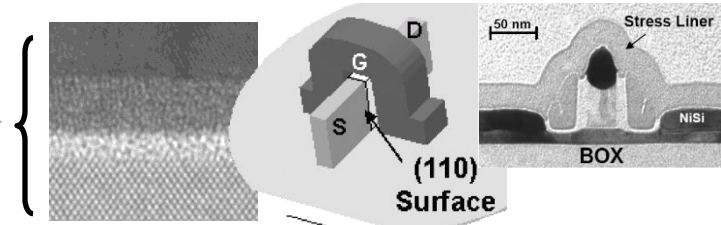
22 nm Tri-Gate transistors provide improved performance at high voltage and an *unprecedented* performance gain at low voltage



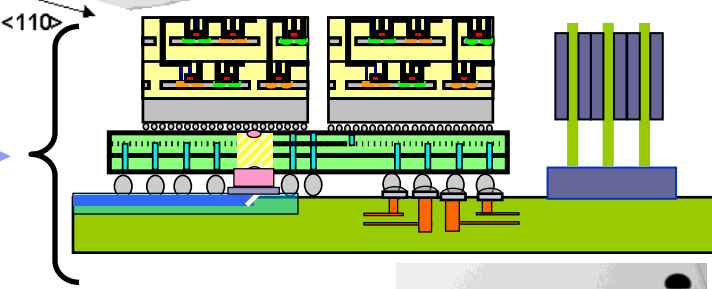
Science and Technology Strategy / Roadmap

2000 2005 2010 2015 2020 2025 2030

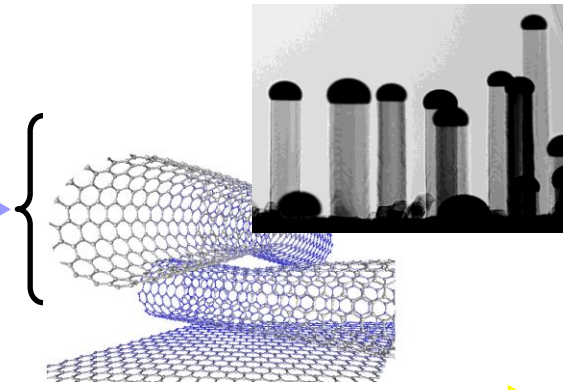
Plan A: Extending Si CMOS



Plan B: Subsystem Integration



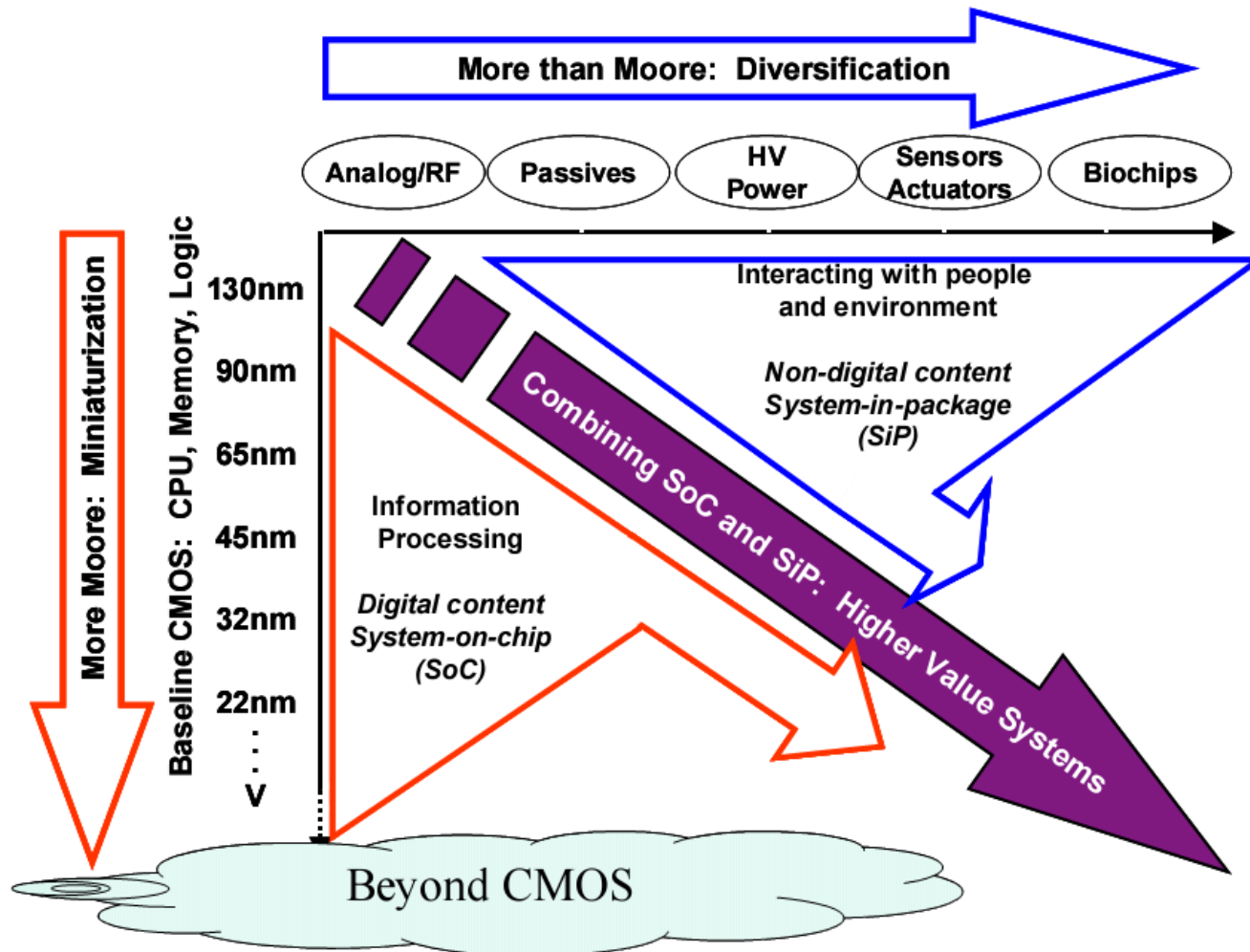
Plan C: Post Si CMOS Options



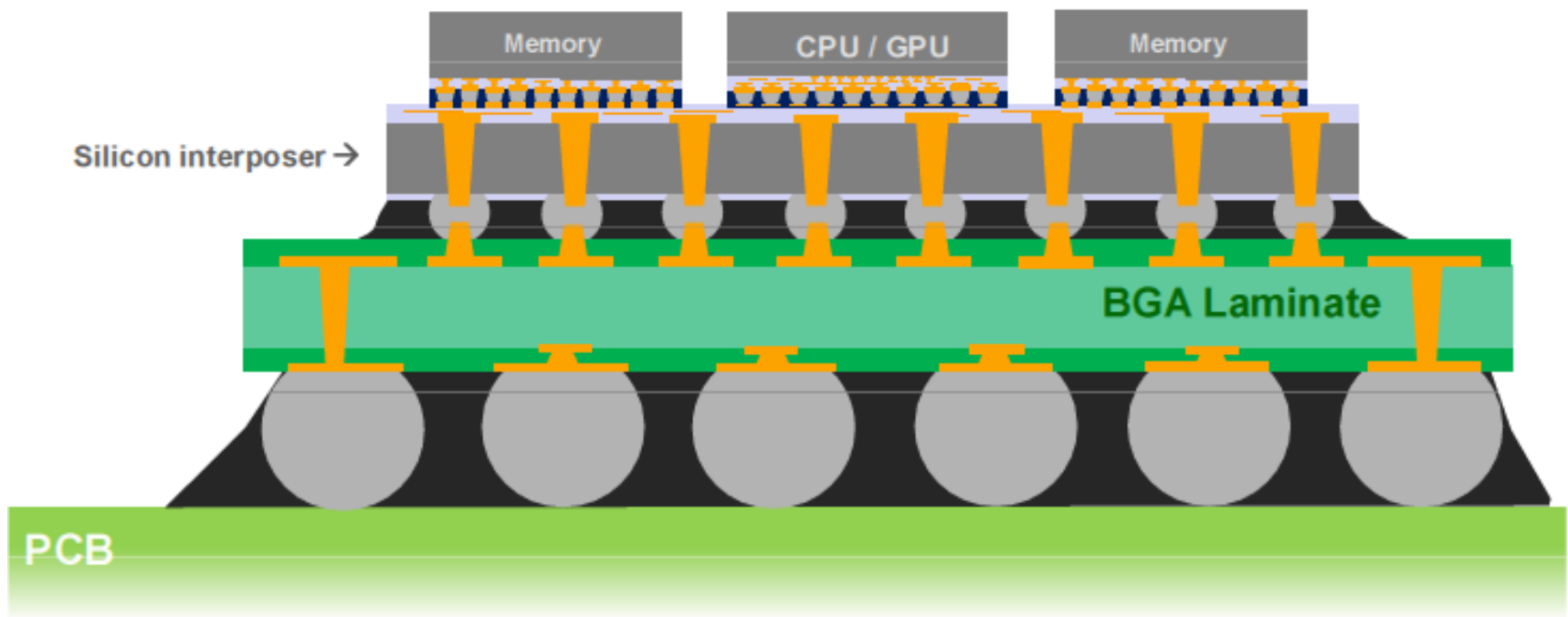
Plan Q: Quantum Computing



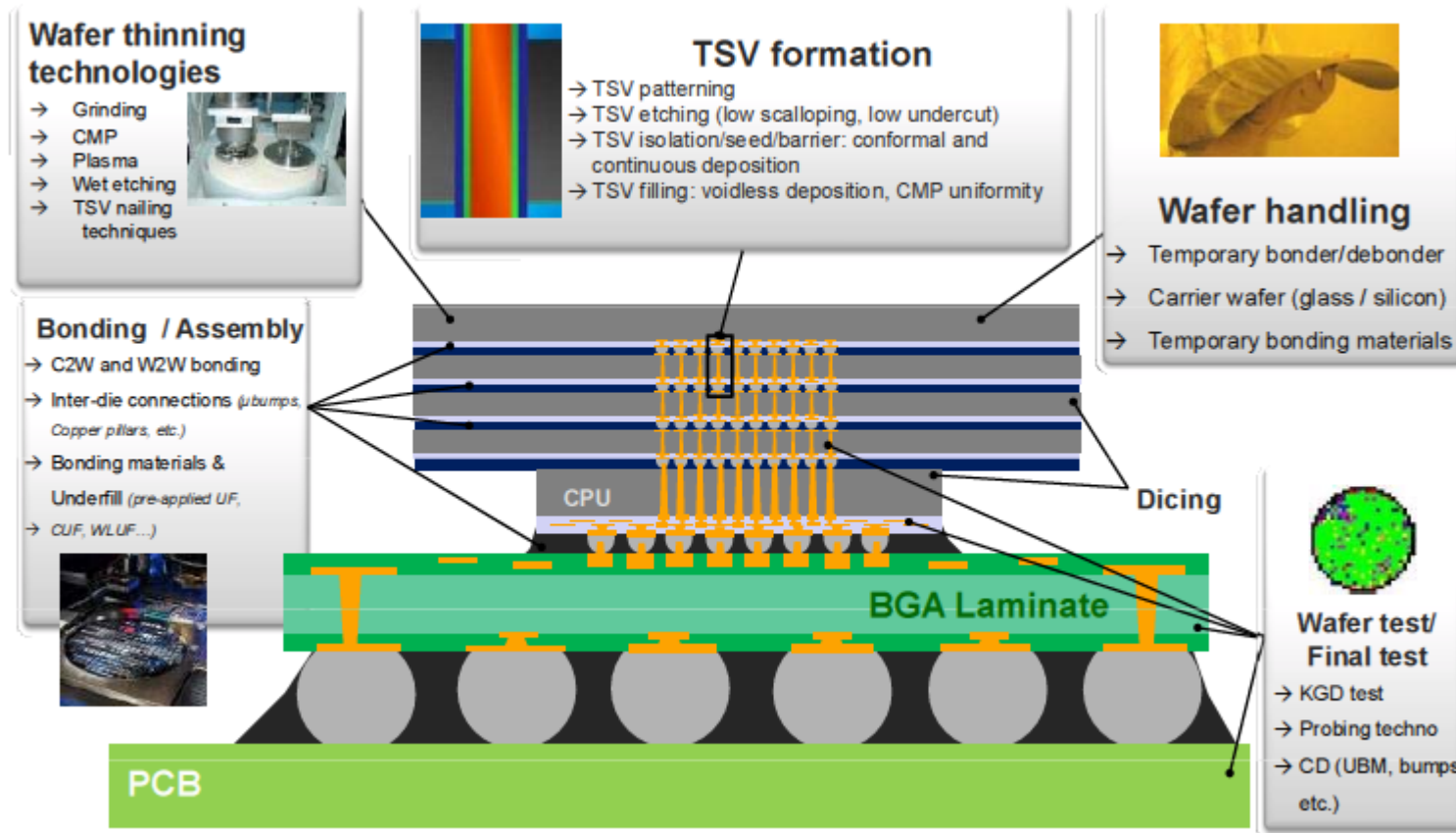
More Moore & More than Moore !!!



2.5D Interposer



3D-IC Technology





DSP Architecture Design?

- Given DSP algorithms, find the “best” solution in the design space under certain constraints
- Or, modified or develop the algorithm to be “hardware oriented” or “hardware friendly,” and then develop the hardware architecture



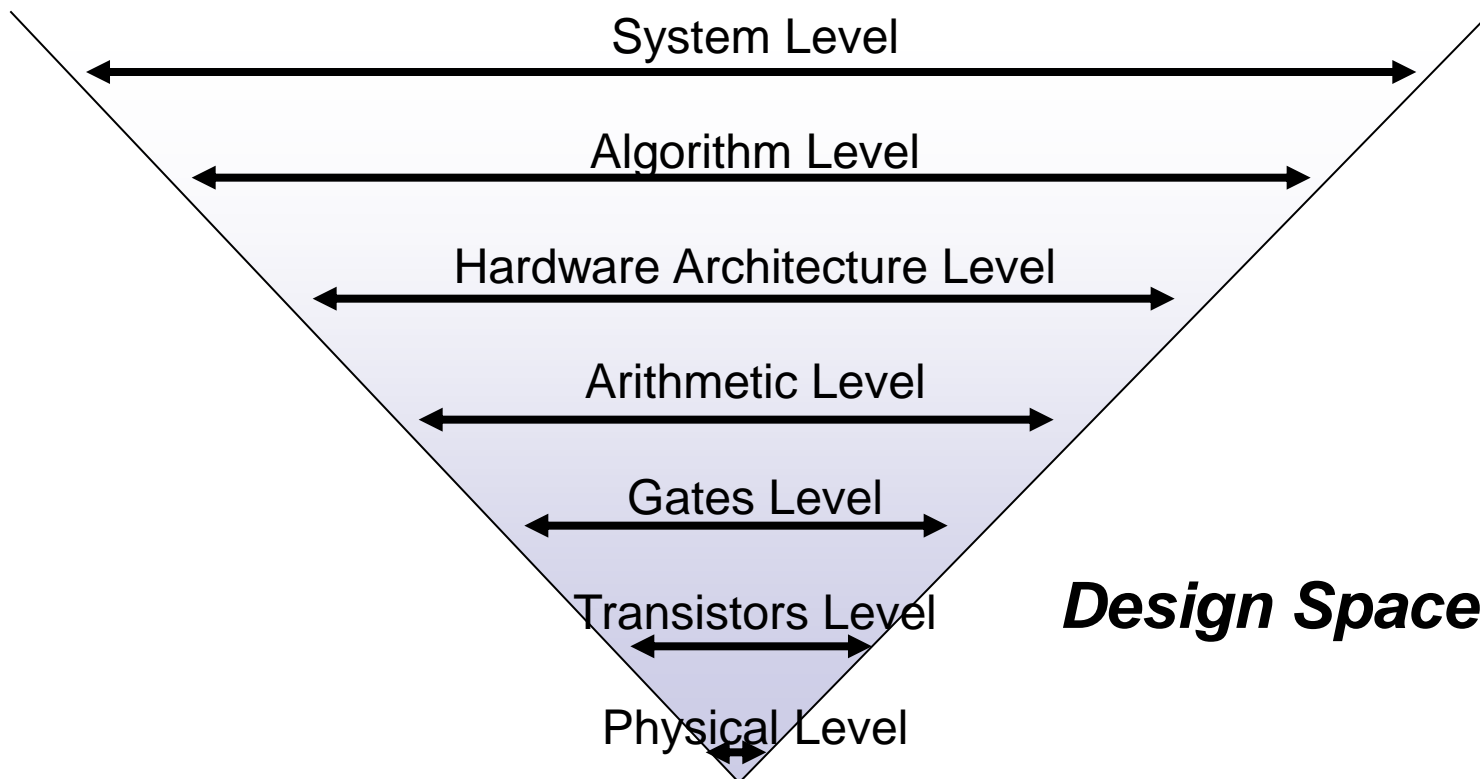
Abstraction Layers

- System (ex: MP3 player)
- Algorithm (ex: FIR filter)
- Hardware architecture (ex: array architecture,...)
- Arithmetic units (ex: multiplier, adder, ...)

- Logic gates (ex: AND, OR, ...)
- Transistors (ex: NMOS, PMOS)
- Layout

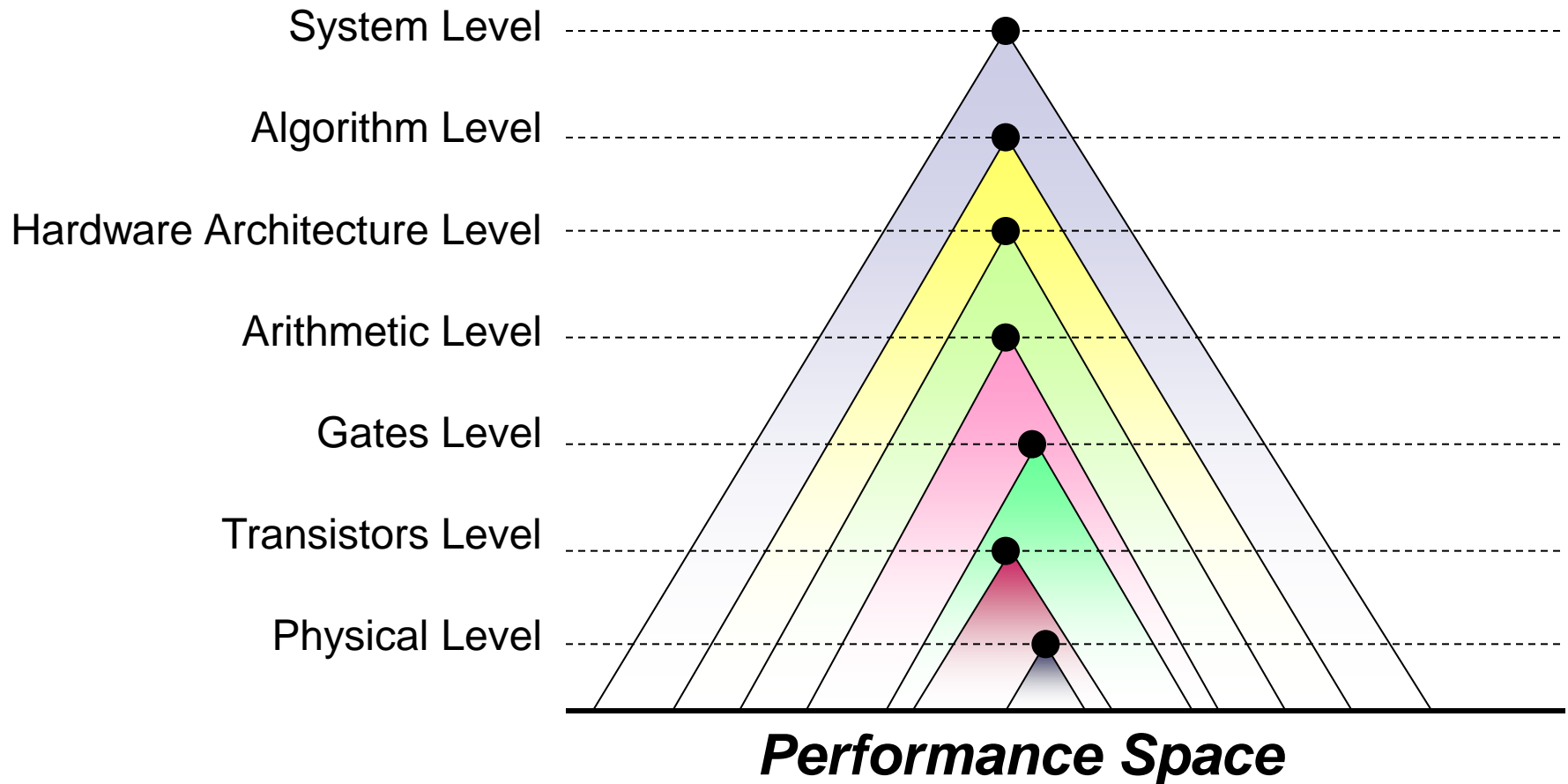


The Higher the Abstraction, The Larger Design Space





The Higher the Abstraction, The More Important





Representations of DSP Algorithms

- DSP algorithms: nonterminating program

$$y(n) = ax(n) + bx(n - 1) + cx(n - 2) \text{ for } n = 1 \text{ to } n = \infty.$$

- Iteration period
- Sampling rate

- Latency
- Throughput
- Clock frequency
- Critical path



Graphical Representations of DSP Algorithms

- Can bridge the gap between algorithmic descriptions and structural implementations
- Block diagram
- Signal-flow graph (SFG)
- Data-flow graph (DFG)
- Dependence graph (DG)

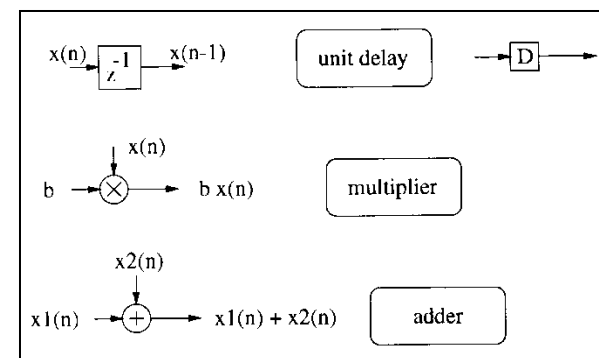
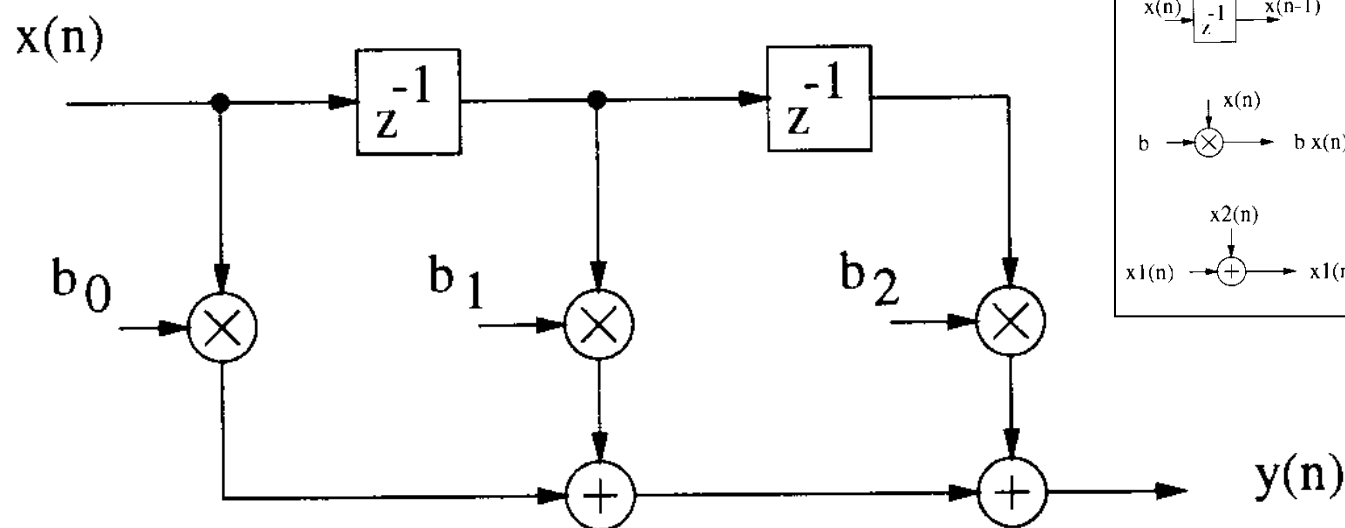


Block Diagram (1/5)

- The most frequently used representation
- Can be constructed with different levels of abstraction
- Can be directly mapped to circuits implementation

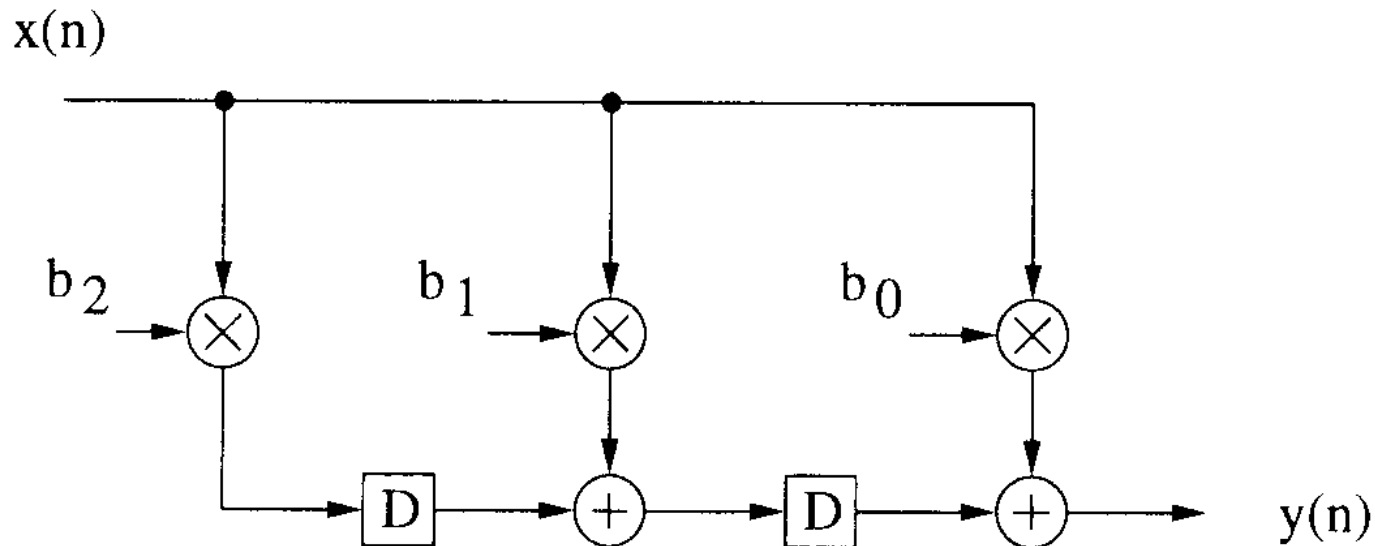
Block Diagram (2/5)

$$y(n] = b_0x(n) + b_1x(n - 1) + b_2x(n - 2)$$



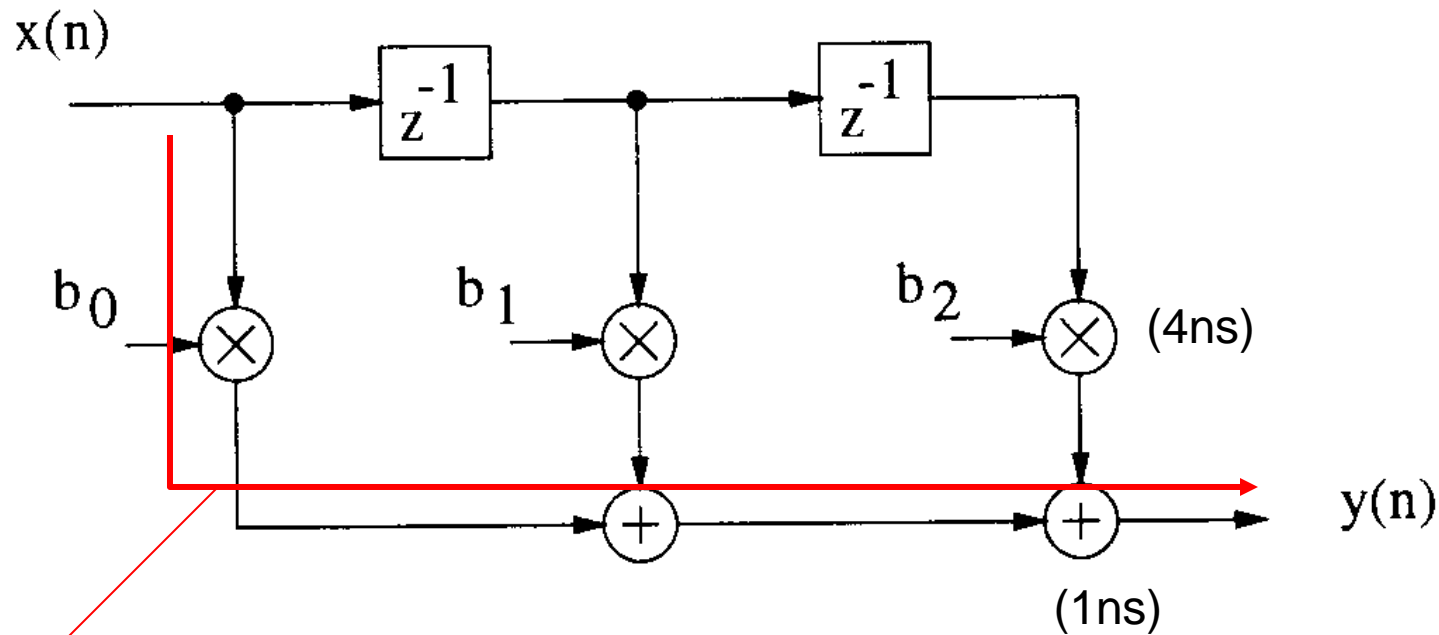
Block Diagram (3/5)

■ Data broadcast FIR filter





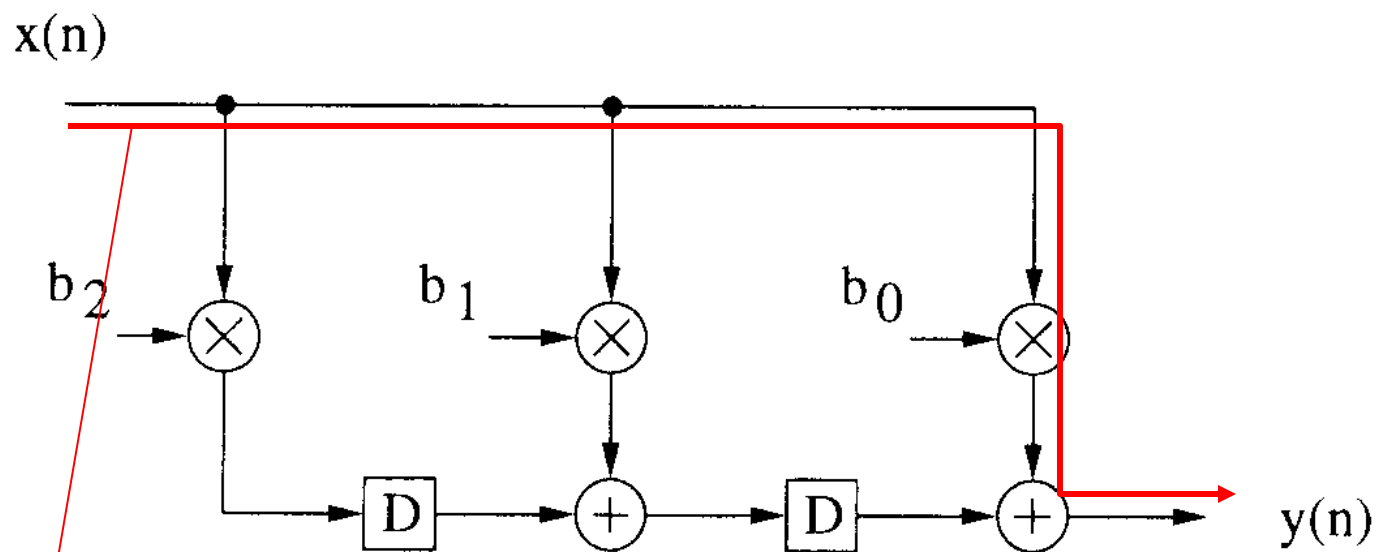
Block Diagram (4/5)



Critical path: $4+1+1=6\text{ns}$

Max clock frequency = $1\text{s}/6\text{ns}=167\text{MHz}$

Block Diagram (5/5)



Critical path: $4+1=5\text{ns}$

Max clock frequency = $1\text{s}/5\text{ns}=200\text{MHz}$

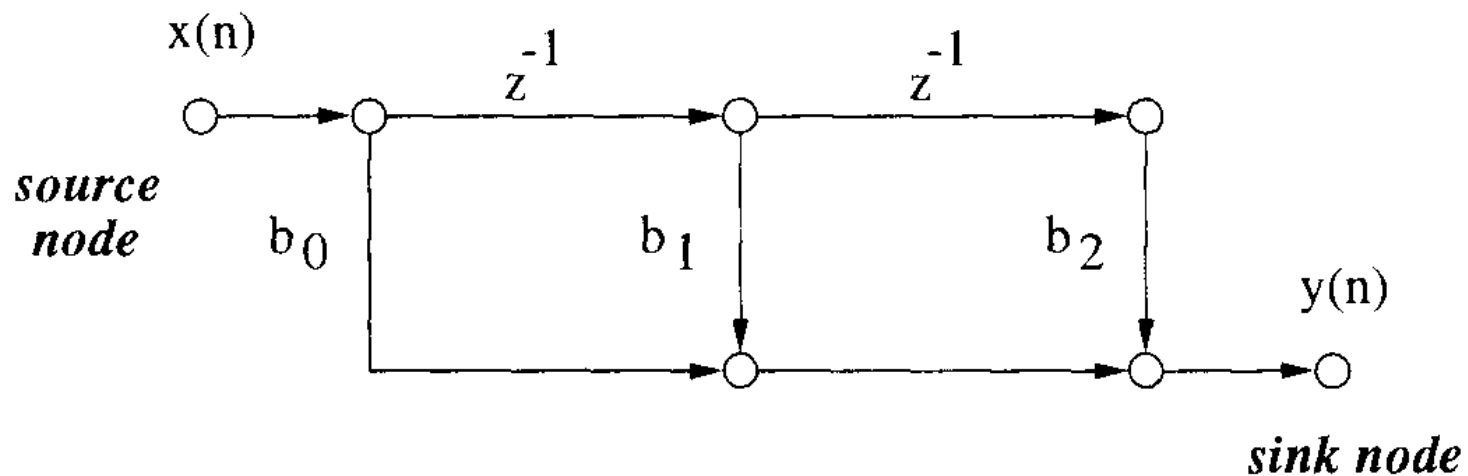


Signal Flow Graph (SFG) (1/4)

- Nodes k
 - Computation or task
- Directed edges (j, k)
 - Linear transformation
 - Source node
 - Sink node

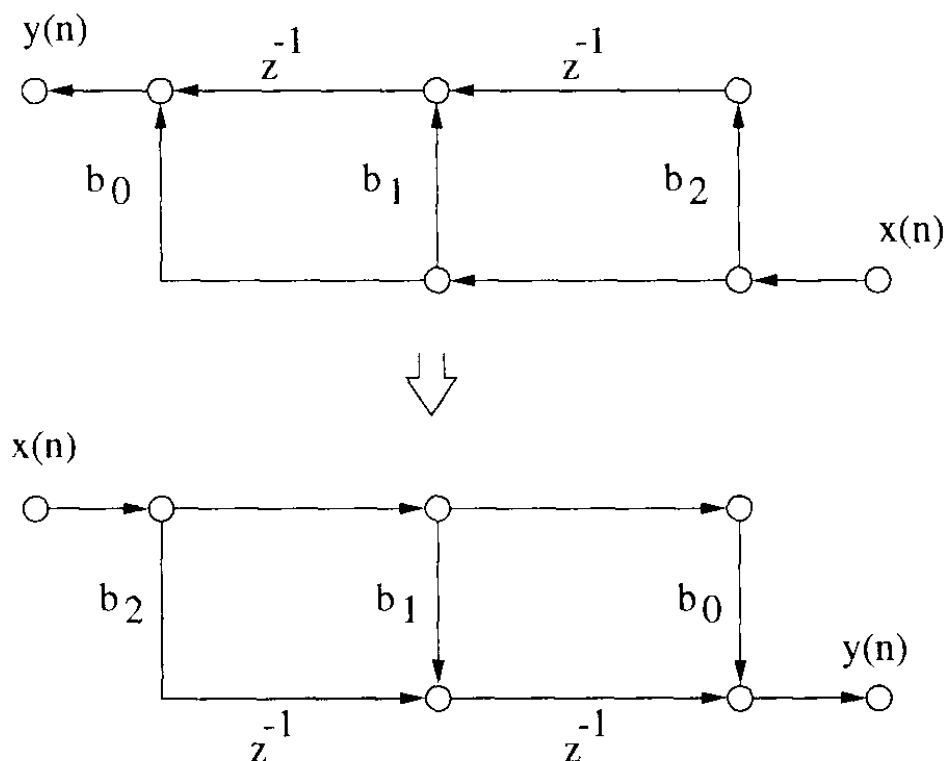


Signal Flow Graph (SFG) (2/4)



Signal Flow Graph (SFG) (3/4)

- Transpose property





Signal Flow Graph (SFG) (4/4)

- Used in digital filter structure and analysis of finite word-length effects
- Only applicable to linear networks
- Cannot be used to describe multi-rate DSP systems



Data-Flow Graph (DFG) (1/4)

- Nodes

- Computations

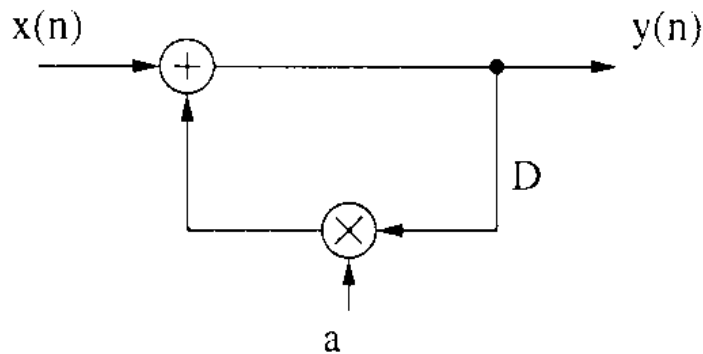
- Directed edges

- Data paths (communication)

- Has a nonnegative number of delays

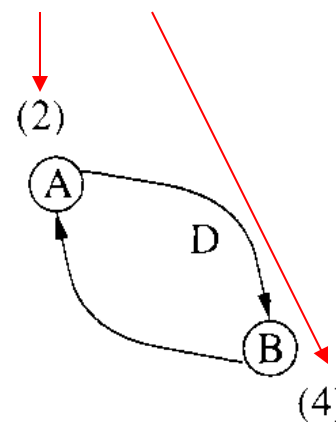


Data-Flow Graph (DFG) (2/4)



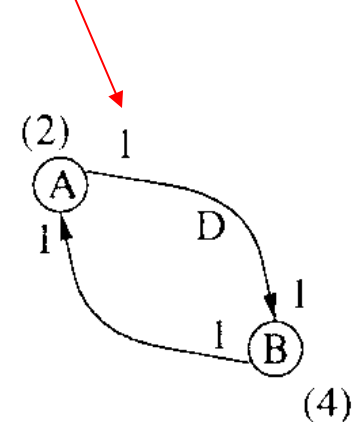
$$y(n) = x(n) + ay(n-1)$$

Execution time



A: +
B: X

Rate



Synchronous DFG

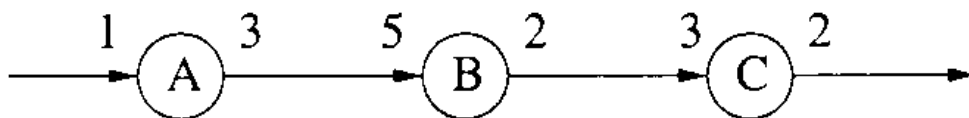


Data-Flow Graph (DFG) (3/4)

- Data-driven property of DSP
 - Any node can fire whenever all the input data are available
 - Intra-iteration precedence constraint
 - Inter-iteration precedence constraint
- Can be used to describe both linear single-rate and nonlinear multi-rate DSP systems

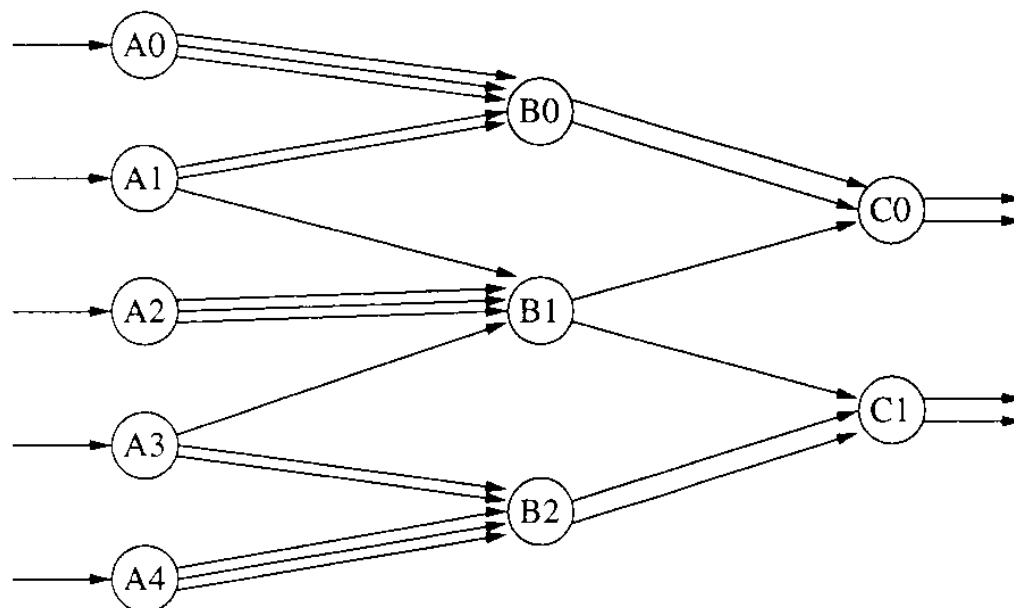
Data-Flow Graph (DFG) (4/4)

- Use single rate DFG (SRDFG) to represent multi-rate DFG (MRDFG)



$$3f_A = 5f_B$$

$$2f_B = 3f_C$$



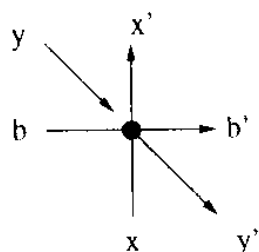
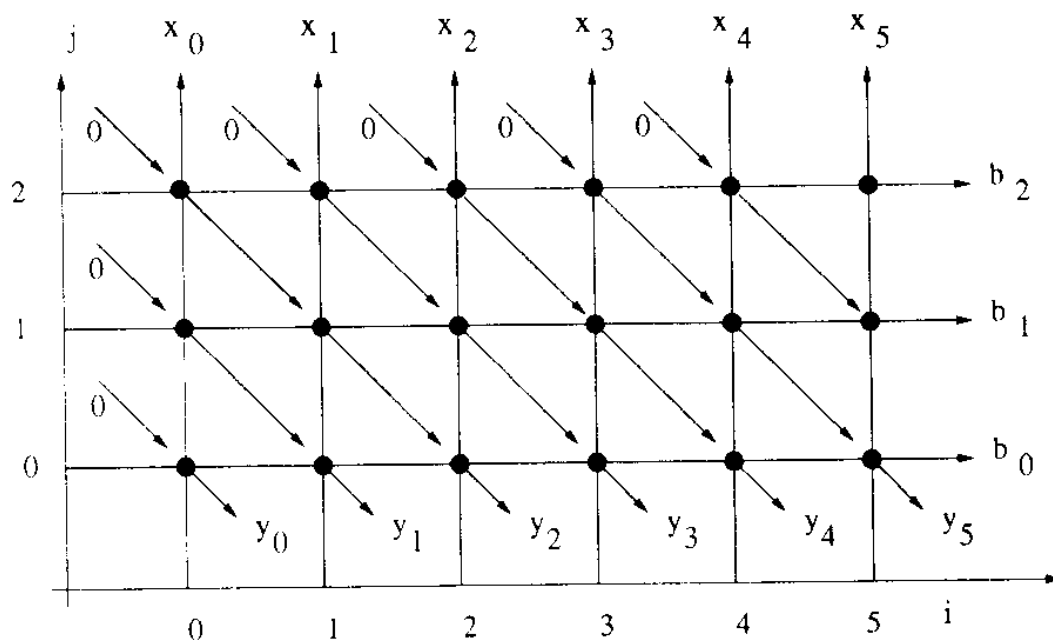


Dependence Graph (1/2)

- A directed graph that shows the dependence of the computation
- Node: computation
- No node in a DG is ever reused on a single computation basis
 - Single-assignment representation
- Used for systolic-array design



Dependence Graph (2/2)



$$\begin{aligned}
 b' &= b \\
 x' &= x \\
 y' &= y + b x
 \end{aligned}$$

$$y(n) = b_0x(n) + b_1x(n - 1) + b_2x(n - 2)$$



DFG v.s. DG

■ DFG

- Nodes only cover computation in one iteration, and will be reused iteratively
- Contain delay elements

■ DG

- Contains computation for all iterations, and is used only once
- No delay elements contained