

A Network Hub Architecture in 2011

David Chinnery, Ben Horowitz
CS 252 Project

1. Introduction

The MESCAL group is investigating architectures required in 2010 for virtual private networks. MESCAL has considered applications and programmability needed at network endpoints and at intermediate nodes. For this project, we consider the architecture for an intermediate node (a network hub).

Projections are for 2011, as that is the nearest year to 2010 on the International Technology Roadmap for Semiconductors (ITRS) [5].

2. Network Model

The maximum network latency is limited by the communication requirements of end users. If the communication is interactive video, then the **maximum acceptable latency is about 0.2 s** [1]. The total latency of end-point and node processing, and time-of-flight must be less than 0.2 s.

Figure 1 depicts the network model, with a packet sent from one end user to another following the lighter path.

2.1. Time of Flight

The delay of light passing through an optic fiber is about 5 ns/m [2]. Repeaters are required to restore the signal about every 100 km, and have a delay of about 0.92 us [2].

For two users communicating from opposite sides of the world, the signal would have to travel about 20,100 km. Thus the worst possible **time of flight delay is about 0.101 s**.

2.2. End-Point Processing

Solutions for end users must be low cost. We assume they have minimal processing capacity to meet the application. The most processing-intensive communication method that might be used by end users is HDTV2, with 1920×1080 frames at 30 Hz. Thus MPEG4 decoding and encoding by end users must have a latency of at most 1/30 s per frame.

MPEG4 breaks the complete video frame up into smaller frames. A 3 million transistor MPEG4 codec uses a frame size of 176×144 and can process at a rate of 10 Hz, with a core speed of 30 MHz [6]. Scaling to 2011, several cores can be used to process HDTV2 frames at 30 Hz: 10 cores if they can run at 0.8 GHz, 5 cores if they can run at 1.5 GHz (see section 3.1 for the scenarios for hardware speed).

2.2.1. 3DES Security

There may be additional end point processing for security. We assume 3DES encryption. If 3DES coding is pipelined with a core like that of [4], additional latency is just for a 176×144 block of the frame.

The maximum bandwidth for MPEG2 (on which MPEG4 is based) is 80 Mbit/s, or 2.7×10^6 bytes/frame [3]. Xentec sells a 7,000 gate 3DES core with a latency of 48 cycles to process a 64 bit word [4], giving 2,000,000 cycles to process an HDTV2 frame. This is substantially less than 1/30 s at processor speeds of 0.8 GHz and above.

Performing 3DES on a 176×144 block takes 24,400 cycles, or about 30 us at 0.8 GHz. This is a negligible addition to the **0.033 s latency of MPEG4 encoding for the HDTV frames**.

2.2.2. Number of Hops

An average packet traveled 15.7 hops on the Internet in 1996 [7]. The average number of hops a packet is the logarithm of the number of Internet nodes [8]. Projecting that the number of Internet nodes in 2011 will be at most the world population of 6.9 billion [9], an average packet will travel 22.7 hops in 2011.

We now consider the maximum number of hops. The worst-case number of nodes traveled by a packet in 1996 was at most 39, with probability 0.999. Linearly extrapolating by the ratio of the average number of nodes, the number of nodes traveled by a packet in 2011 will be 56.3, with probability 0.999.

2.2.3. Transceiver Latency

With 56 nodes, there are 110 transceivers (each end user has only one transceiver). The delay of a fiber optic transceiver is 50 ns [2], giving a total delay for all the transceivers of 5.5 us, which is negligible.

2.3. Latency Implications

Totaling the latency for two end users, and time-of-flight, this leaves **0.032 s processing latency total for the nodes**. The **allowable latency per node is 0.00057 s**, and we assume the same latency for the gateways.

3. Hub Architecture

Each routing chip has limited processing bandwidth. However, it is straightforward for multiple chips to be

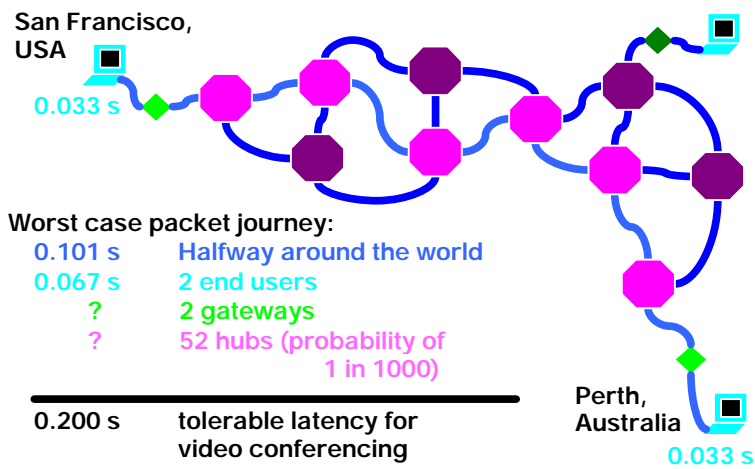


Figure 1. Illustration of packet journey from one end point, out through a gateway, through hubs, then out through a gateway to another end point user.

used in parallel for terabyte optical links connecting to a hub, sending a segment of wavelengths to each chip.

3.1. Chip Frequency

A fundamental limitation on both the I/O bandwidth and processing speed of hub routing chips is the clock frequency at which they can be run. According to the ITRS, an ASIC can be expected to run at 1.5 GHz (across chip and locally for ASIC logic) in 2011 [5].

A more conservative scenario extrapolates the clock frequency of a core processing unit from 1999 to 2011. The Intel IXP1200 is an ASIC used for routing today, with a speed of 0.166 GHz in a 0.28 μm process [10]. Scaling this linearly to 0.18 μm (in reality, the speed will scale slightly sub-linearly), gives a speed of 0.258 GHz in 0.18 μm . From 0.18 μm to 0.05 μm in 2011 [5], the ITRS predicts that ASIC speeds will change from 0.5 GHz to 1.5 GHz. Scaling further according to the ITRS, this would conservatively predict that the IXP1200 could be run at 0.775 GHz in 2011.

We assume that cores available today can run at the clock frequency in the respective scenario.

3.2. Chip I/O Bandwidth

In 2011 the ITRS predicts that there will be 927 chip-to-package pins, which can operate at both 0.775 GHz and 1.5 GHz (there are other pins that can provide power to the chip) [5]. The majority of these pins must be used to provide I/O bandwidth (it turns out to be the bottleneck), but we suppose that 32 pins would be reserved for other I/O, such as updating the routing congestion table for determining the next hop for a packet (see Figure 2).

This leaves 448 pins (rounding up) dedicated for input, and 448 pins dedicated for output, as we stream data across the chip. At a frequency of 0.775 GHz, the maximum bandwidth is 347 Gbit/s. At a frequency of 1.5 GHz, the maximum bandwidth is 672 Gbit/s.

3.3. IXP1200 Processors in Hub Chips

The Intel IXP1200 has 6.5×10^6 transistors [10]. From the ITRS, an 8 cm^2 chip of maximum area in 2011 will have 6.5×10^9 transistors [5]. Conservatively, we suppose that 70% of the chip area is required for wiring, memory and control circuitry, allowing 300 IXP1200 processors to be included on a hub routing chip in 2011.

From simulations by Scott Weber and Fernando De Bernardinis, an IXP1200 can process 2.3×10^6 packets of 64 byte length in 1999. Scaling, an IXP1200 can process 10.7×10^6 packets at 0.775 GHz, or 20.8×10^6 packets at 1.5 GHz. Thus a hub routing chip in 2011 could process 3.21×10^9 packets at 0.775 GHz, or 6.24×10^9 packets at 1.5 GHz.

The minimum possible packet length is 20 bytes, the header of an IPv4 packet. At maximum bandwidth, if all the packets were of 20 bytes length, there would be 2.17×10^9 packets/s at a frequency of 0.775 GHz, or 4.2×10^9 packets/s at a frequency of 1.5 GHz. Thus with 300 IXP1200 processors on the chip, the maximum bandwidth can be processed easily, even if all the packets carry no information - the bottleneck is the I/O bandwidth of these routing chips.

3.4. Hub Chip Architecture

The processing units and buses proposed for our routing chip are shown in Figure 2. A hub router must achieve a very high throughput of packets to their next destination. Limited other functionality is required. In MESCAL, we allow programmability for different network protocols and different configurations.

From the 56 byte input bus, the queue in controller Q_{in} control determines which queue packets go to for processing, based on packet priority and loading of queues (high priority packets can be sent to queues with free IXP 1200 processors).

The twenty in queues, with fifteen IXP 1200s each, allow several levels of packet priorities, where several queues can be reserved for processing high priority packets. Alternatively, the bus could be split into buses for packets from several different prior hops (this would require additional functionality within the header detection unit, which has not been considered here). The in queues assign the arriving packet headers to available IXP 1200s for processing to determine the next destination, and can send the packet body on to the out queues.

After the IXP 1200 has processed the packet header, the packet can be sent, and the out queue signals the out queue controller Q_{out} control. The out queue controller

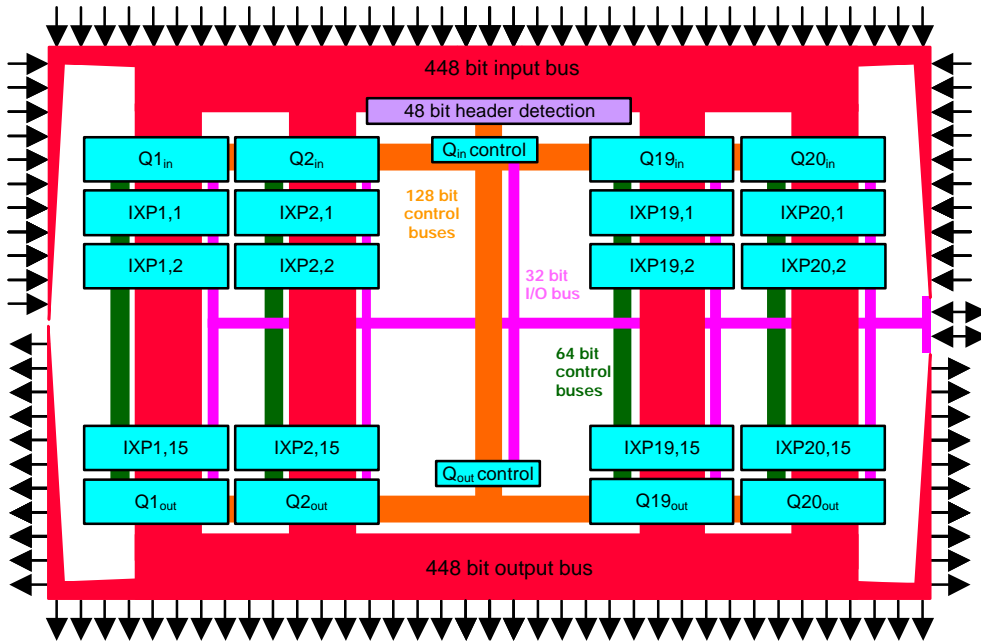


Figure 2. Overview of hub chip processing units and buses

arbitrates which queue can send a packet to the bus, based on queue priorities.

The in queues $Q_{k_{in}}$, out queues $Q_{k_{out}}$, and queue controllers can be processors with significant programmability. Estimating 20 million transistors (not including local memory to store packets) for each of these processors on the chip, they would take 840 million transistors, or 13% of the silicon area - leaving 57% of the chip area for buses and memory, which is more than sufficient.

3.5. Header Detection

In order to stream data across the chip rapidly and avoid processing congestion within individual queues, the number of packets being processed by a queue must be up-to-date for the in queue controllers. This requires rapid packet header detection, to detect new packets arriving. We have chosen 48 bits of header detection: this is sufficient to match at Ethernet destination address (6 bytes); and far more than the 4 bits in an IPv4 or IPv6 packet header used for packet detection. This allows different network protocols to be used, programming the mask for header detection.

While the rest of the chip runs at ASIC speeds, for rapid header detection, a custom macro would be used, running at about 10 GHz (roadmap local clock speed for custom [5]). Within the main clock cycle, there are 13 cycles for the header detection unit at 0.775 GHz, or 6

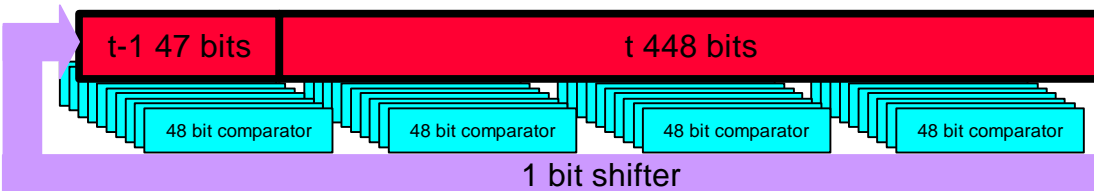


Figure 3. Packet header detection

cycles at 1.5 GHz. One cycle is required to send the header position of any packets in the 448 bits, the remaining cycles can be used for comparing the incoming bits with the header mask, shifting the bits to detect for the header in a different position on the next cycle. To scan all 448 bits, 38 comparators are required at 0.775 GHz main clock speed (each comparator can detect for the header in 12 positions within the main clock cycle), or 90 comparators at 1.5 GHz main clock speed (each comparator can detect for the header in 5 positions within the main clock cycle). Figure 3 shows how the shifter and 48 bit comparators test all positions of the incoming data each cycle (note that a one bit shifter can be achieved simply with pass gates and wiring shifted one bit back to the registers).

Figure 4 shows a single bit unit of the 48-bit comparator, comparing the mask to the input bit. If the value of this particular bit doesn't matter (a "don't care") then the output will be 1. This 10-transistor unit computes $care_i \wedge (input_i \oplus mask_i)$, taking advantage of both polarities of signals being available from registers to give a compact cell.

Each comparator also has a counter to set the position a header is detected if it is found. Within 56 bytes, as the minimum packet size is 20 bytes, at most 3 headers could be detected. For simplicity, the header positions can be sent with 31 bits: splitting the 448 bits up into three groups of 128 bits and a fourth of 64 bits, and using an additional bit signal for each of these four groups to indicate whether or not a header was found in that range.

Accounting for the counters, comparators and registers, about 30,000 transistors are required for the header detection macro in the 0.775 GHz scenario, and 71,000 transistors in the 1.5 GHz scenario. The header detection macro takes very little area on the chip, which can have up to about 6.5 billion transistors in 2011 [5].

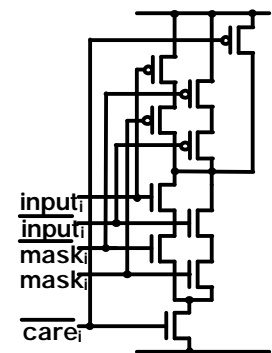


Figure 4. Header detection comparator base cell, comparing one unit of the input with the mask.

3.6. Hub Chip Memory

In IXP 1200 simulations, Scott Weber and Fernando De Bernardinis used 4 megabytes of DRAM and 2 megabytes of SRAM. SRAM takes four times the area of DRAM [12], and DRAM is predicted to have a density of 7.51 Gbit/cm² in 2011 [5]. For 300 IXP 1200s, integrating the memory on chip, the memory area taken up per IXP 1200 is 0.41 cm². (Note that memory speeds in 2011 scale sufficiently fast from 1999, to perform at the same or better rates relative to the main clock speed.)

Additional memory is required to store packets in the queues. As high throughput is required, with rapid access to packets within the queues once they are scheduled to send a packet out, we assume this is register memory, which takes ten times the area of DRAM [13]. Conservatively assuming that each queue must be able to store a packet of maximum size,¹ where the maximum size of an IPv6 packet is 65,536 bytes [14], the 0.021 Gbit of registers take an area of 0.014 cm².

The total area for the memory of the hub chips is 0.42 cm², 5% of the total chip area of 8 cm². Additional memory may be desirable for the queue controller and queue processors, depending on any additional tasks required, such as routing hash table construction based on real time routing congestion statistics.

3.7. Hub Chip Power Consumption

We compare the power consumption of the suggested hub chip, relative to today's processors. According to the ITRS, 90 W of power can be dissipated in 1999 compared with 174 W of power in 2011, and supply voltages are 1.8 V and 0.6 V respectively for high performance chips [15].

Dynamic power dissipation through a capacitor, with capacitance C and voltage V across it, at a switching frequency f , is given by $P = fCV^2$ [16]. In 0.18 μm , clock frequencies of 1.2 GHz are expected [5]. Thus in 1999 the dynamic power dissipation is given by $P = 3.89C$, and the dynamic power dissipation is given by $P = 0.28C$ in the 2011 scenario with a clock speed of 0.775 GHz. Comparing the 1999 and 2011 scenarios, and accounting for the additional power that can be dissipated by chip packaging (a factor of $\times 1.93$), the chip capacitance could be 26.8 times larger and still meet packaging power dissipation limitations in 2011.

The predicted chip size of 8 cm² in 2011 is double that of 4 cm² chips available today [5]. The transistor density will increase from $20 \times 10^6/\text{cm}^2$ to $811 \times 10^6/\text{cm}^2$ [5]. The capacitance of a transistor is proportional to Ae/t_{ox} [16], where the gate oxide thickness t_{ox} decreases

¹ This is only necessary for robustness when faced with severe network congestion at other nodes- in general, such a large packet would be being sent off chip before the packet body had finished arrived.

proportional to the voltage (constant field), the area A decreases as the square of the decrease in transistor channel length (0.18 μm to 0.05 μm). We will assume the dielectric permittivity of the oxide remains the same (assuming silicon dioxide used as the gate oxide in 1999 and 2011). Thus the capacitance per cm² increases by a factor of $\times 9.4$ and the chip capacitance increases by $\times 18.8$ (chip size is double). Comparing the increase in chip capacitance, with the factor it could increase by above, the chip will be within packaging power dissipation limits.²

4. Simulation

We have written a discrete-event simulator that models our processor. Though our simulator is in the style of Ptolemy II, we estimated that it would be more efficient to write our own. The simulator is written in Java.

Each of the components of the processor is modeled as a Java object. The components consist of the in and out busses, the in and out queue controllers, the 40 in and out queues, and the 300 IXP 1200s. The components communicate by sending each other messages. The in bus receives packets and notifies the in queue controller of their arrival. The in queue controller sends messages to the most lightly loaded in-queue, instructing that queue to read a packet. The in queue then sends header information to its most lightly loaded IXP 1200, requesting that the IXP 1200 compute the next hop information. The IXP 1200 computes the next hop information, which it sends to its out queue. Whenever the out bus is idle and a packet needs to be shipped, the out queue controller chooses the out queue with the heaviest load, and instructs that out queue to ship.

The sizes of packets are obtained by sampling from random variables. In particular, we have modeled three scenarios: (1) the sizes of packets are distributed according to [11]; (2) the sizes of packets are distributed according to a conservative approximation of [11]; and (3) the size of packets is 20 bytes (i.e., as small as possible). The last scenario represents the worstcase load for our router.

We model the time required by a processor to compute the next hop information for a packet according to two scenarios: (a) the number of cycles is given by a Gaussian random variable with mean 340 and standard deviation 130 (from the simulations by Weber and De Bernardinis: the minimum, maximum, and average processing times were 75, 600, and 345 cycles, resp.); and (b) the number of cycles is always 600. Scenario

² This is not the case if the chip speed is 1.5 GHz, according to these simple calculations we have made here - the acceptable increase in capacitance would be only 13.8 times larger. However, as we show in the simulation results, the actual chip size can be smaller, as far fewer processors are required in realistic scenarios to meet bandwidth processing requirements.

Scenario	Scenario Description	# of IXP 1200s	Max. packets/processor	Max. packets/queue
(1b)	(size from [11], cycles = 600)	300	1	7
		20	7	7
(2b)	(size = approx. of [11], cycles = 600)	300	1	11
		40	6	11
		20	10	10
(3b)	(size = 20, cycles = 600)	300	7	91
(1a)	(packet size from [11], cycles = Gaussian)	300	1	5
		20	5	5
(2a)	(size = approx. of [11], cycles = Gaussian)	300	1	6
		20	7	7
(3a)	(size = 20, cycles = Gaussian)	300	4	54

Table 1. Queue and IXP 1200 maximum loading for different packet size and processing time distributions

(b) represents the worst case for our router. The results of our simulation are summarized in Table 1.

We have designed the simulator to be as flexible as possible. The parameters of our chip - including the number of queues, number of IXP 1200s, and the statistical models - may easily be varied. Further, the simulations are independent of chip frequency, as processing times are in cycles.

Each IXP 1200 has six micro-engines to calculate the next destination of packets, and context switches between packets when performing memory operations required in hash table lookups to find the next destination. Thus, an IXP 1200 can process more than six packets at the same time. Our simulations show that our chip is easily able to handle even the worstcase load of scenario (3b). In this case, the maximum number of packets being processed by each IXP1200 processor is 7.

4.1. Simulation Conclusions

The latency of a packet to get onto the chip, have the header processed, and arrive at the out queue ready to be sent off is at most 605 cycles, providing the IXP 1200s are not over-loaded. This gives a processing time per node of 0.78 us in the 0.775 GHz scenario, and 0.40 us in the 1.5 GHz scenario. When the packet is ready to be sent, the worst possible scenario is that a maximum length packet has been arriving and started to be sent while the packet was being processed, requiring 1170 cycles to send the 65,536 byte packet through the 56 byte output bus - this would take 1.51 us in the 0.775 GHz scenario, and 0.78 us in the 1.5 GHz scenario. Thus, the maximum latency for a packet per hub node is 2.4 us in the 0.775 GHz scenario, and 1.3 us in the 1.5 GHz scenario, meeting latency requirements even in the worst-case simulation scenario (3b). As I/O bandwidth is the bottleneck, the silicon area of the chip could be reduced, decreasing the cost of the chip, while still meeting processing requirements. Scenarios with fewer IXP 1200s, but still twenty queues, are also shown in Table 1. As illustrated in the realistic scenarios of (1a) and (2a) in Table 1, 20 IXP 1200s would be sufficient to process the packets with no additional latency as 7 or fewer packets are being processed by an IXP 1200 at

any time. The chip size could be reduced from 8 cm², down to about 0.5 cm². However, the hub router would be susceptible to denial-of-service attacks causing congestion by sending many 20 byte packets (simulation scenario (1a)).

5. Acknowledgements

Michael Shilman, Mel Tsai and Trevor Meyerowitz outlined an initial MESCAL network model, and provided several of the references. Thanks to Scott Weber and Fernando De Bernardinis for information on IXP1200 simulations.

6. References

- [1] Michael Katz, "When CTI Meets the Internet," Telecommunications Online, July 1997. <http://www.telecomsmag.com/issues/199707/tcs/katz.html>
- [2] Telebyte, *Model 372 - 100Base-TX to Fiber Optic Transceiver Reference Manual*. <http://www.telebyteusa.com/catalog/manuals/m372.htm>
- [3] *MPEG-2 Frequently Asked Questions List*. <http://www.unix.digital.com/demos/freetkit/docs/mpeg2/FAQ>
- [4] Xentec, *X_3DES Triple DES*. http://www.xentec-inc.com/X-datasheets/x_3DES.PDF
- [5] *International Technology Roadmap for Semiconductors, 1999 Edition, Overall Roadmap Technology Characteristics and Glossary*. http://www.itrs.net/1999_SIA_Roadmap/ORTC.pdf
- [6] M. Takahashi et al., "A 60mW MPEG4 Video Codec Using Clustered Voltage Scaling With Variable Supply-Voltage Scheme," ISSCC, 1998.
- [7] National Laboratory for Applied Network Research, *Assessing average hop count of a wide area Internet packet*. <http://www.nlanr.net/NA/Learn/wingspan.html>
- [8] Larry Roberts, "Beyond Moore's Law: Internet Growth Trends," IEEE Computer Magazine, January 2000, 117-119.

- [9] U.S. Bureau of the Census, International Data Base, *Total Midyear Population for the World: 1950-2050*.
<http://www.census.gov/ipc/www/worldpop.html>
- [10] Tom R. Halfhill, "Intel Network Processor Targets Routers," Microprocessor Report, September 13, 1999, vol. 13-12.
- [11] National Laboratory for Applied Network Research, *WAN packet size distribution*, June 1997.
<http://www.nlanr.net/NA/Learn/packetsizes.html>
- [12] *What is every kind of RAM?* February 2000.
<http://whatis.com/ramguide.htm#kindsof>
- [13] Michael Flynn, *EE382 Processor Design: Silicon Area and Cost Models*. 1999.
<http://www.stanford.edu/class/ee382/areacost/ppframe.htm>
- [14] Carl Erickson, *Data Communications Lecture Notes: IP*
<http://www.csd.uu.se/~carle/datakomm/Notes/IP/tsld021.htm>
- [15] *International Technology Roadmap for Semiconductors, 1999 Edition, Assembly & Packaging*.
http://www.itrs.net/1999_SIA_Roadmap/Assembly.pdf
- [16] Neil H. E. Weste, and Kamran Eshraghian, *Principles of CMOS VLSI Design*, 2nd ed. Addison-Wesley, 1992.