Chapter 2. The Basics

Problem 2.1

A four segment pipeline implements a function and has the following delays for each segment: ($b = .2$)

Segment #	Maximum delay
1	17
2	15
3	19
4	14

Where $c = 2\,\text{ns}$,

a. What is the cycle time that maximizes performance without allocating multiple cycles to a segment?

   Since the maximum stage delay is 19 ns, this is the shortest possible delay time that does not require multiple cycles for any given stage. Thus, $19 + 2 = 21\,\text{ns}$ is the minimum cycle time for this pipeline.

b. What is the total time to execute the function through all stages?

   There are four stages, each of which is clocked at 21 ns. $4 \times 21 = 84\,\text{ns}$ is thus the total delay (latency) through the pipeline.

c. $S_{\text{opt}} = \sqrt{\frac{(1-0.2)(17+15+19+14)}{(0.2)(2)}} \approx 11.$

   Let $T_{\text{seg}} = 7\,\text{ns}$

   $S = 11$

   $G = \frac{1}{1+10\times.2}\frac{1}{7+2} = 37.0$ MIPS

   Let $T_{\text{seg}} = 7.5\,\text{ns}$

   $S = 10$

   $G = \frac{1}{1+9\times.2}\frac{1}{7.5+2} = 37.6$ MIPS

   Let $T_{\text{seg}} = 8.5\,\text{ns}$

   $S = 9$

   $G = \frac{1}{1+8\times.2}\frac{1}{8.5+2} = 36.6$ MIPS

   Let $T_{\text{seg}} = 9.5\,\text{ns}$

   $S = 8$

   $G = \frac{1}{1+7\times.2}\frac{1}{9.5+2} = 36.2$ MIPS

   The cycle time that maximizes performance $= 7.5 + 2 = 9.5\,\text{ns}$

Problem 2.2

Repeat problem 1 if there is a 1 ns clock skew (uncertainty of $\pm 1\,\text{ns}$) in the arrival of each clock pulse.

a. What is the minimum cycle time without allocating multiple cycles to a segment?

Since the maximum stage delay is 21 ns, the minimum cycle time should be $21 + 2 = 23$ ns.

b. What is the total time to execute the function through all stages?

There are four stages, each of which is clocked at 23 ns. $4 \times 23 = 92$ ns is thus the total delay (latency) through the pipeline.

c. Now $c = 4$.

$$S_{\text{opt}} \sqrt{\frac{(1 - .2)(76)}{(.2)4}} = 8 \text{ or } 9$$

Use 8:

$$\text{Cycle time} = \frac{7.6}{8} + 4 = 13.5 \text{ ns}.$$

Problem 2.3

a. No uncontrolled clock skew allowed

Minimum clock cycle time = largest $P_{\max} + C = 16 + 3 = 19$ ns

Latency = 4 (19) = 76 ns

b. Wave pipelining allowed

$\Delta T_{\min}$ = largest $(P_{\max} - P_{\min}) + C = 7 + 3 = 10$ ns

Latency = $(14 + 3) + (12 + 3) + (16 + 3) + (11 + 3) = 65$ ns

$CS_1 \uparrow = 17 \mod 10 = 7$ ns (or -3 ns)
$CS_2 \uparrow = 32 \mod 10 = 2$ ns
$CS_3 \uparrow = 51 \mod 10 = 1$ ns
$CS_4 \uparrow = 65 \mod 10 = 5$ ns

Problem 2.5

$T = 120$ ns, $C = 5$ ns, $b = 0.2$, $S_{\text{opt}} = \sqrt{\frac{(1-b)(1+k)T}{bC}}$.

k	S_{opt}
0.05	10.04
0.08	10.18
0.11	10.32
0.14	10.46
0.17	10.60
0.20	10.73

Optimum segments vs. overhead is shown in figure 1.

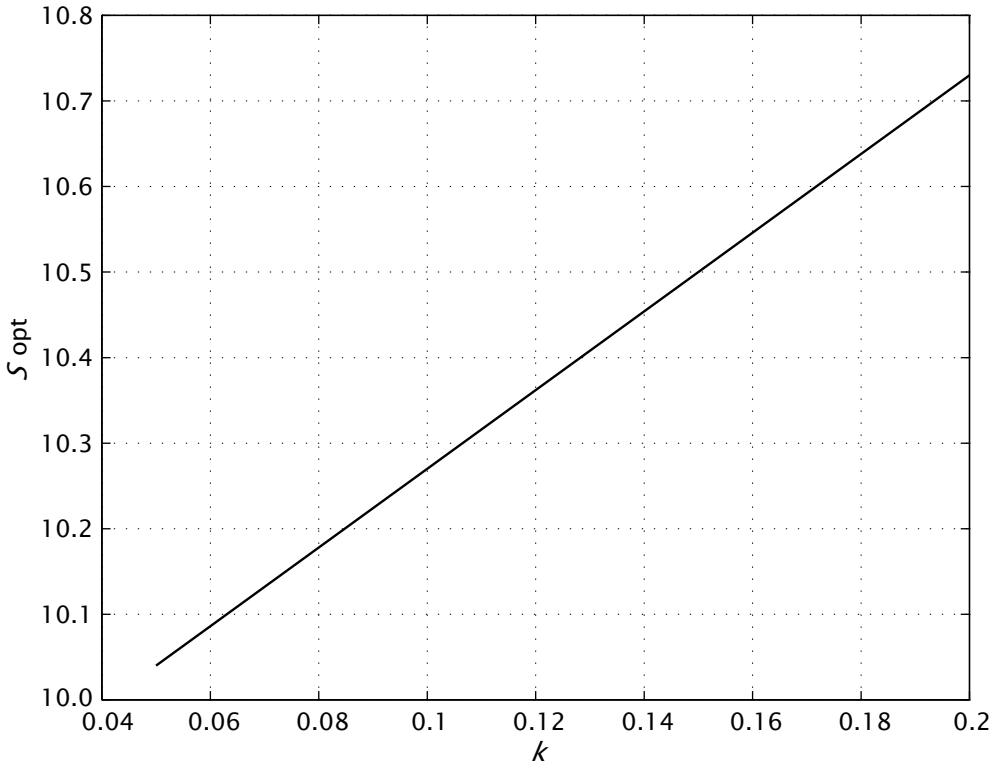

Figure 1: Problem 2-5.

Problem 2.6

a. $G = \frac{1}{(1+b(S-a))(C+T(1+k)/S)}$

Let $\frac{dG}{dS} = 0$

$-b(CS^2 + T(1+k)S) + T(1+k)(1+b(S-a)) = 0$

$-bCS^2 + T(1-ba)(1+k) = 0$

$S_{\text{opt}} = \sqrt{\frac{(1-ba)(1+k)T}{bC}}$

Problem 2.7

a. It is relatively easy to see how a clock skew of δ increases the clocking overhead in a traditionally clocked system by 2δ. Assume we have two latches and each of them is controlled by clock 1 and clock 2. Then, it is possible for clock 1 for the first latch to tick late by δ, and clock 2 for the second latch to tick early by δ. In this case, the cycle time is increased by 2δ.

b. It is a little harder to show that ΔT = largest $(P_{\max} - P_{\min})$ + oldC + 4δ. Once again, uncertainty in the arrival of data determines the clock cycle. Both the clock before and the clock after the stage are affected by the uncontrollable clock skew.

Clock 2 after the stage clocks the data out of the stage. This must tick before $P_{\min}$ and after $P_{\max}$ (plus setup and delay time). Because of its uncertainty, it must be moved d earlier than $P_{\min}$ and δ later than $P_{\max}$.

The clock 1 before the stage causes additional uncertainty in $P_{\min}$ and $P_{\max}$. (You can think of $P_{\min}$ decreasing and $P_{\max}$ increasing.) Thus, it must be moved an additional δ earlier and a δ later.

Problem 2.8

Delay $R \rightarrow$ALU$\rightarrow R = 16$ ns (assume it breaks into 5-5-6 ns)

Instruction or data cache data access = 8 ns

a. Compute S_{opt}, assuming b = .2

$T = 4 + 6 + 8 + 3 + 12 + 9 + 3 + 6 + 8 + 3 + 16 + 2 = 80$ ns

$k = .05$

$C = 4$ ns

$S_{\text{opt}} = \sqrt{\frac{(1-b)(1+K)T}{bC}} = \sqrt{\frac{.8 \times 1.05 \times 80}{.2 \times 4}} = 9.16$, round to 9

b. Partition into approximately S_{opt} segments

(i) First partitioning:

Stage	Actions	Time
IF1	PC→MAR (4), cache dir (6)	10
IF2	cache data (8)	8
ID1	data transfer to IR (3),decode (6)	9
ID2	decode (6)	6
AG	address generate (9)	9
DF1	address→MAR (3), cache dir (6)	9
DF2	cache data access (8)	8
EX1	data→ALU (3),execute (5)	8
EX2	execute (5)	5
PA	execute (6), put away (2)	8

$T_{\text{seg}} = 10$ ns

10 stages

$\Delta T = (1 + K)T_{\text{seg}} + C = (1.05)(10\,\text{ns}) + 4\,\text{ns} = 14.5$ ns

$T_{\text{inst}} = \text{stages} \times \Delta T = 145$ ns

(ii) Second partitioning:

Stage	Actions	Time
IF1	PC→MAR (4), cache dir (6)	10
IF2	cache data (8)	8
ID1	data transfer to IR (3),decode (6)	9
ID2	decode (6)	6
AG	address generate (9)	9
DF1	address→MAR (3), cache dir (6)	9
DF2	cache data access (8),data→ALU (3)	11
EX	execute (10)	10
PA	execute (6)put away (2)	8

$T_{\text{seg}} = 11$ ns

9 stages

$\Delta T = (1 + K)T_{\text{seg}} + C = (1.05)(11\,\text{ns}) + 4\,\text{ns} = 15.55$ ns

$T_{\text{inst}} = \text{stages} \times \Delta T = 139.95$ ns

(iii) Third partitioning:

Stage	Actions	Time
IF1	PC→MAR (4), cache dir (6)	10
IF2	cache data (8), data transfer (3)	11
ID	instruction decode (12)	12
AG	address generate (9), MAR (3)	12
DF1	cache dir (6)	6
DF2	cache data (8),data→ALU (3)	11
EX1	execute (10)	10
EX2	execute (6),put away (2)	8

$T_{\text{seg}} = 12\,\text{ns}$

8 stages

$\Delta T = (1+K)T_{\text{seg}} + C = (1.05)(12\,\text{ns}) + 4\,\text{ns} = 16.6\,\text{ns}$

$T_{\text{inst}} = \text{stages} \times \Delta T = 8(16.6) = 132.8\,\text{ns}$

(iv) Fourth partitioning:

Stage	Actions	Time
IF1	PC→MAR (4), cache dir +data (14)	18
ID	data transfer (3), decode (12)	15
AG	AG (9), MAR (3), cache dir (6)	18
DF2	cache data (8),data→ALU (3)	11
PA	execute (16),put away (2)	18

$T_{\text{seg}} = 18\,\text{ns}$

5 stages

$\Delta T = (1+K)T_{\text{seg}} + C = (1.05)(18\,\text{ns}) + 4\,\text{ns} = 22.9\,\text{ns}$

$T_{\text{inst}} = \text{stages} \times \Delta T = 5(22.9) = 114.5\,\text{ns}$

c. Performance

- $G = \frac{1}{1+(S-1)b} \times \frac{1}{(1+k)(T_{\text{seg}})+C}$
- $b = .2$

$G(T_{\text{seg}} = 10\,\text{ns}) = \frac{1}{1+9\times.2} \times \frac{1}{14.5\,\text{ns}} = 24.6$ MIPS

$G(T_{\text{seg}} = 11\,\text{ns}) = \frac{1}{1+8\times.2} \times \frac{1}{15.55\,\text{ns}} = 24.7$ MIPS

$G(T_{\text{seg}} = 12\,\text{ns}) = \frac{1}{1+7\times.2} \times \frac{1}{16.6\,\text{ns}} = 25.1$ MIPS

$G(T_{\text{seg}} = 18\,\text{ns}) = \frac{1}{1+4\times.2} \times \frac{1}{22.9\,\text{ns}} = 24.3$ MIPS

Of the pipelines considered here, with the above assumptions, the 8-stage ($T_{\text{seg}} = 12\,\text{ns}$) pipeline has the best performance. However, the 5-stage pipeline also has decent performance and would be easier to implement.

Problem 2.9

$b = .25$, $C = 2\,\text{ns}$, $k = 0$.

Function	Delay
A	6
B	8
C	3
D	7
E	9
F	5

$T = 6 + 8 + 3 + 7 + 9 + 5 = 38$.

a. Optimum number of pipeline segments, ignoring quantization

$S_{\text{opt}} = \sqrt{\frac{(1-.25)(1+0)38}{.25 \times 2}} = 7.55$, round down to 7.

b. Cycle time

Since $S_{\text{opt}} = 7$, one function unit can be split into 2 stages. Split function unit E since it has the longest delay.

$\Delta T = 8 + 2 = 10\,\text{ns}$

c. Performance

$G = \frac{1}{1+(S-1)b} \times \frac{1}{T_{\text{seg}}+C} = \frac{1}{1+6\times.25}\frac{1}{8+2\text{ ns}} = 40$ MIPS.

Performance $= \frac{1}{1+(7-1)0.25} = 0.4$ inst/cycle.

d. Can you find better performance?

Assume that each function unit cannot be merged with neighboring function units.

A 6
B 8 Can be divided into 4, 4 ns stages
C 3
D 7 Can be divided into 3.5, 3.5 ns stages
E 9 Can be divided into 4.5, 4.5 ns stages
F 5

(i) Do not split any function units.
$T_{\text{seg}} = 9\,\text{ns}$
$S = 6$
$G = \frac{1}{1+5\times.25}\frac{1}{9+2\text{ ns}} = 40.4$ MIPS

(ii) Split function units B and E.
$T_{\text{seg}} = 7\,\text{ns}$
$S = 8$
$G = \frac{1}{1+7\times.25}\frac{1}{7+2\text{ ns}} = 40.4$ MIPS

(iii) Split function units B, D and E.
$T_{\text{seg}} = 6\,\text{ns}$
$S = 9$
$G = \frac{1}{1+8\times.25}\frac{1}{6+2\text{ ns}} = 41.7$ MIPS
$T_{\text{seg}} = 6\,\text{ns}$ gives the best performance.

e. The adjusted cycle time $= 10 + 2(1) = 12\,\text{ns}$.

Problem 2.10

a. Without aspect-mismatch adjustment,

- Bits per line = 256 bits/line
- Total number of lines = 32KB/32B = 1024 lines
- Tag bits = 20

Data bits = $32\,\text{KB} \times 8$ bits per byte $= 32 \times 1024 \times 8 = 262,144$ bits

Tag bits = 1024 lines $\times$ 20 bits per byte = 20480 bits

Cache size = $195 + .6(262,144 + 20,480) = 169,769.4$ **rbe**

b. With aspect-mismatch adjustment

With aspect-ratio mismatch, 10% of our area is wasted, so the actual area which is taken up is only 90% of the total area. We divide by .9 to find the corrected area,

$$\text{Area} = \frac{169,769.4}{.9} = 188,632.7 \textbf{ rbe}$$

Problem 2.11

$A = (1.4 \text{ cm})^2 = 1.96 \text{ cm}^2$

a. 6-inch wafer = 15.24-cm wafer

$N = \frac{\pi}{4A}(d - \sqrt{A})^2 = \frac{\pi}{4 \times 1.4^2}(15.24 - 1.4)^2 \approx 76$

Year	ρ_D	Y	N_G	Wafer cost	Effective die cost
1	1.5	.053	4	\$5000	\$1250
2	1.325	.074	5	\$4625	\$925
3	1.15	.105	7	\$4250	\$607
4	.975	.148	11	\$3875	\$352.30
5	.8	.21	16	\$3500	\$218.70

b. 8-inch wafer = 20.32-cm wafer

$N = \frac{\pi}{4A}(d - \sqrt{A})^2 = \frac{\pi}{4 \times 1.4^2}(20.32 - 1.4)^2 \approx 143$

Year	ρ_D	Y	N_G	Wafer cost	Effective die cost
1	1.5	.053	7	\$10000	\$1428.60
2	1.325	.074	10	\$9125	\$912.50
3	1.15	.105	15	\$8250	\$550.00
4	.975	.148	21	\$7375	\$351.20
5	.8	.21	30	\$6500	\$216.70

It is more effective to use an 8-inch wafer since the effective die cost is lower for the last 4 years.

Problem 2.12

$A = -\frac{\ln(\text{yield})}{\rho_D} = -\frac{\ln(.1)}{1} = 2.3 \text{ cm}^2$

20% of die area is reserved for pads, etc., so we have left = 1.84 cm^2

10% of die area is reserved for sense amps, etc., so we have left = 1.61 cm^2

$f = 1\mu m = 2\lambda$

Cell size = $135\lambda^2 = 33.75\,\mu m^2$

Capacity = $\frac{1.61}{33.75\times10^{-8}} = 4.770 \times 10^6$ bits

Of this, we can only use 4 Mb = 2^{22} = 4,194,304 bits

New cell area = 1.61 $cm^2 \times \frac{4194304}{4.770\times10^6} = 1.42$ cm^2 which is 70% of die area.

Total die area = 1.42 $cm^2 \times \frac{100}{70} = 2.03$ cm^2

New Yield = $e^{-1\times2.03} = 13.1$ %

(We can also assume that the 10% overhead for sense amps, etc. is calculated from the total cell area. In this case, the capacity is 4.9478×10^6 bits. The new total die area = 1.95 cm^2 and the new yield = 14.2 %)

Problem 2.13

$f = 1\mu m = 2\lambda$

Cell size = $135\lambda^2 = 33.75\,\mu m^2$

Total cell area = $1024 \times 1024 \times 33.75\,\mu m^2 = 0.354$ cm^2

Total die area = $0.354 \times \frac{100}{70} = 0.506$ cm^2

Yield = $e^{-1\times0.506} = 0.603$

$N = \frac{\pi}{4A}(d - \sqrt{A})^2 = \frac{\pi}{4\times0.506}(21 - \sqrt{0.506})^2 \approx 638$

$N_G = 638 \times 0.603 \approx 384$

Cost of a $1^M \times 1^b$ die = $\frac{\$5000}{384} \approx \$13$

(Can also assume that the 10% overhead for sense amps, etc. is calculated from the total cell area. In this case, the total die area = 0.487 cm^2 and the cost ≈ $12.30)

Problem 2.14

$A = 2.3$ cm^2

Yield (.5 defects/cm^2) = $e^{-\rho_D \times A} = e^{-.5\times2.3} = 31.7\%$

Yield (1 defects/cm^2) = $e^{-\rho_D \times A} = e^{-1\times2.3} = 10.0\%$

Die (15 cm) = $\frac{\pi}{4A}(d - \sqrt{A})^2 = \frac{\pi}{4\times2.3} \times (15 - 1.52)^2 = 62$

Die (20 cm) = $\frac{\pi}{4A}(d - \sqrt{A})^2 = \frac{\pi}{4\times2.3} \times (20 - 1.52)^2 = 116$

$N_G(d = 15, \rho_D = .5) = 62 \times .317 = 19.7$, cost/good die = $\frac{\$5000}{19.7} = \$254$

$N_G(d = 15, \rho_D = 1) = 62 \times .100 = 6.2$, cost/good die = $\frac{\$5000}{6.2} = \$806$

$N_G(d = 20, \rho_D = .5) = 116 \times .317 = 36.8$, cost/good die = $\frac{\$8000}{36.8} = \$217$

$N_G(d = 20, \rho_D = 1) = 116 \times .100 = 11.6$, cost/good die = $\frac{\$8000}{11.6} = \$690$

For either defect density, the larger wafer is more cost effective for the given die size and wafer costs.

Problem 2.15

a. Using one chip:

$A = 280 \text{ mm}^2 \times \frac{10}{8} = 350 \text{ mm}^2 = 3.5 \text{ cm}^2$

$N = \frac{\pi}{4 \times 3.5}(15 - \sqrt{3.5})^2 = 38.68$

$Y = e^{-\rho_D A} = e^{-1 \times 3.5} = .03$

$N_G = N \cdot Y = 38.68 \times .03 = 1.16 \approx 1$

Effective cost = \$4000

Cost for one processor = \$4000 + \$50 = \$4050

b. Using two chips:

$A = 1.75 \text{ cm}^2$

$N = \frac{\pi}{4 \times 1.75}(15 - \sqrt{1.75})^2 = 83.95$

$Y = e^{-\rho_D A} = e^{-1 \times 1.75} = .174$

$N_G = N \cdot Y = 83.95 \times .174 = 14.6 \approx 14$

Effective cost = $\frac{\$4000}{14} = \$285.70$

Cost for one processor = $\$285.70 \times 2 + \$50 \times 2 = \$671.40$

c. Using four chips:

$A = .875 \text{ cm}^2$

$N = \frac{\pi}{4 \times .875}(15 - \sqrt{.875})^2 = 177.56$

$Y = e^{-\rho_D A} = e^{-1 \times .875} = .417$

$N_G = N \cdot Y = 177.56 \times .417 \approx 74$

Effective cost = $\frac{\$4000}{74} = \$54.05$

Cost for one processor = $\$54.05 \times 4 + \$50 \times 4 = \$416.20$

d. Using eight chips:

$A = .4375 \text{ cm}^2$

$N = \frac{\pi}{4 \times .4375}(15 - \sqrt{.4375})^2 = 369$

$Y = e^{-\rho_D A} = e^{-1 \times .4375} = .646$

$N_G = N \cdot Y = 369 \times .646 \approx 238$

Effective cost = $\frac{\$4000}{238} = \$16.80$

Cost for one processor = $\$16.80 \times 8 + \$50 \times 8 = \$534.40$

Using four chips shows the smallest cost per processor.