Computer Architecture Recitation 1 – Part A

Prepared by

Kadir Özlem & Fırat Öncel

Question 1- Pipeline

- We designed a pipeline P to perform the task **T** on the elements of an array.
- The duration for executing the task on an array with **two elements** using the pipeline P is $\mathbf{T}_2$ = 150 ns.
- The duration for executing the task on an array with **three elements** using the pipeline P is $\mathbf{T}_3$ = 180 ns

a) Calculate the number of stages (layers) of the pipeline P (**k**=?).

b) The duration for executing the task on one element without a pipeline is $\mathbf{t}_n$ = 90ns.
Calculate the speedup achieved by the pipeline P if the array has an infinite number of elements ($S_{n\to\infty}$ = ?).

c) We design a new pipeline $\mathbf{P}_{\mathbf{new}}$ for the same task **T**. $\mathbf{P}_{\mathbf{new}}$ has more stages than the previous pipeline P given above ($k_{new} > k$).
The duration for executing the task on only the first element using the pipeline $\mathbf{P}_{\mathbf{new}}$ is $\mathbf{T}_{new1}$ = 120 ns.
Compare the speedups of the pipelines P and P_{new} for an array with infinite number of elements.
($S_{new\infty} > S_\infty$, $S_{new\infty} < S_\infty$, or $S_{new\infty} = S_\infty$)? Explain briefly.

Solution 1

- **a)** *Calculate the number of stages (layers) of the pipeline P (**k**=?).*
- T_2 = 150 ns
- T_3 = 180 ns
- $T_3 - T_2 = \mathbf{t_p = 30\ ns}$ Period of clock cycle
- $T_2 = (k+1)\ t_p \rightarrow \mathbf{k = 4}$ Number of layers

- **b)** *The duration for executing the task on one element without a pipeline is* $\mathbf{t}_n$ *= 90ns.*
- *Calculate the speedup achieved by the pipeline P if the array has an infinite number of elements (*$S_{n\rightarrow\infty}$ *= ?).*
- $S_{n\rightarrow\infty} = \frac{t_n}{t_p} = \frac{90}{30}$
- $\boldsymbol{S_{n\rightarrow\infty} = 3}$

Solution 1

- ***c)*** *We design a new pipeline* ***P_{new}*** *for the same task* ***T****. P_{new} has more stages than the previous pipeline P given above*
- *($k_{new} > k$).*
- *The duration for executing the task on only the first element using the pipeline P_{new} is* ***T_{new1}*** *= 120 ns.*
- *Compare the speedups of the pipelines P and P_{new} for an array with infinite number of elements.*
- *($S_{new\infty} > S_\infty$, $S_{new\infty} < S_\infty$, or $S_{new\infty} = S_\infty$)? Explain briefly.*

- $T_1 = k * t_p = 120\ ns$
- $T_{new1} = k_{new} * t_{pnew} = 120\ ns$
- $T_1 = T_{new1}\ \&\ k_{new} = k \Rightarrow\ t_{pnew} < t_p$
- *The clock cycle of the new pipeline is shorter.*
- $S_{new\infty} = \frac{t_n}{t_{pnew}}\ \&\ S_\infty = \frac{t_n}{t_p}\ \&\ t_{pnew} < t_p \Rightarrow\ S_{new\infty} > S_\infty$

Question 2

A RISC CPU has an instruction pipeline with the following 5 stages:

IF: Instruction fetch

DR: Instruction Decode, Read registers

EX: Execute

ME: Memory

WB: Write back

The CPU does not include any forwarding (by-pass) connections. The CPU writes data to registers in the first half of the cycle (rising edge) and reads data from registers in the second half of the cycle (falling edge). Branch target address calculation is performed in the EX stage end result is sent directly to the IF stage.

INSTRUCTION SET:

LDL	X(Rs), Rd	Rd ← M[Rs + X]	Load
STL	X(Rs), Rm	M[Rs + X] ← Rm	Store
ADD	Ri, Rj, Rd	Rd ← Ri + Rj	Add
SUB	Ri, Rj, Rd	Rd ← Ri - Rj	Subtrack
BRU	Y	PC ← PC + Y	Branch relative

```
        SUB   R2, R2, R1
        ADD   R1, 20, R2
        LDL   $100(R2), R3
        ADD   R3, R4, R5
        SUB   R6, R7, R6
        STL   $04(R2), R6
        LDL   $08(R2), R5
        BRU   FINISH
        ADD   R5, R6, R5
        SUB   R1, R5, R3
FINISH: ADD   R2, 4, R2
```

a) Draw the space-time diagram for the execution of the given program, in the given instruction pipeline. Solve all data and branch conflicts using NOOP instructions. What is the total amount of penalty in clock cycles caused by conflicts for the given piece of code?

b) To minimize the amount of penalty, apply the optimized software-based solutions to the conflicts, if it is possible. Remember; the results generated by the program cannot be changed. What is the total amount of penalty in clock cycles with the new solutions?

Solution 2

Solution A (50 Points)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SUB R2, R2, R1	IF	DR	EX	ME	WB																		
NOOP		IF	DR	EX	ME	WB																	
NOOP			IF	DR	EX	ME	WB																
ADD R1, 20, R2				IF	DR	EX	ME	WB															
NOOP					IF	DR	EX	ME	WB														
NOOP						IF	DR	EX	ME	WB													
LDL $100(R2), R3							IF	DR	EX	ME	WB												
NOOP								IF	DR	EX	ME	WB											
NOOP									IF	DR	EX	ME	WB										
ADD R3, R4, R5										IF	DR	EX	ME	WB									
SUB R6, R7, R6											IF	DR	EX	ME	WB								
NOOP												IF	DR	EX	ME	WB							
NOOP													IF	DR	EX	ME	WB						
STL $04(R2), R6														IF	DR	EX	ME	WB					
LDL $08(R2), R5															IF	DR	EX	ME	WB				
BRU FINISH																IF	DR	EX	ME	WB			
NOOP																	IF	DR	EX	ME	WB		
NOOP																		IF	DR	EX	ME	WB	
FINISH: ADD R2, 4, R2																			IF	DR	EX	ME	WB

Total amount of penalty is 10 clock cycles.

Solution 2

Solution B (50 Points)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SUB R2, R2, R1	IF	DR	EX	ME	WB											
NOOP		IF	DR	EX	ME	WB										
NOOP			IF	DR	EX	ME	WB									
ADD R1, 20, R2				IF	DR	EX	ME	WB								
SUB R6, R7, R6					IF	DR	EX	ME	WB							
NOOP						IF	DR	EX	ME	WB						
LDL $100(R2), R3							IF	DR	EX	ME	WB					
STL $04(R2), R6								IF	DR	EX	ME	WB				
BRU FINISH									IF	DR	EX	ME	WB			
ADD R3, R4, R5										IF	DR	EX	ME	WB		
LDL $08(R2), R5											IF	DR	EX	ME	WB	
FINISH: ADD R2, 4, R2												IF	DR	EX	ME	WB

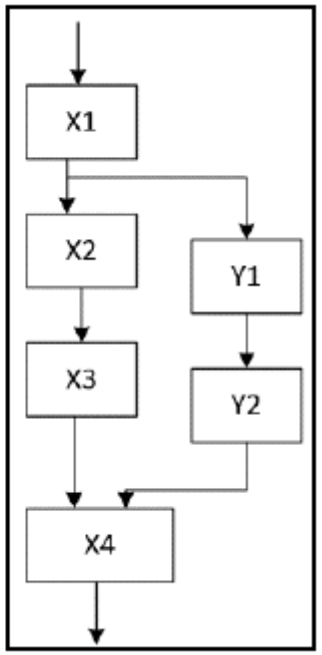
Total amount of penalty is 3 clock cycles.

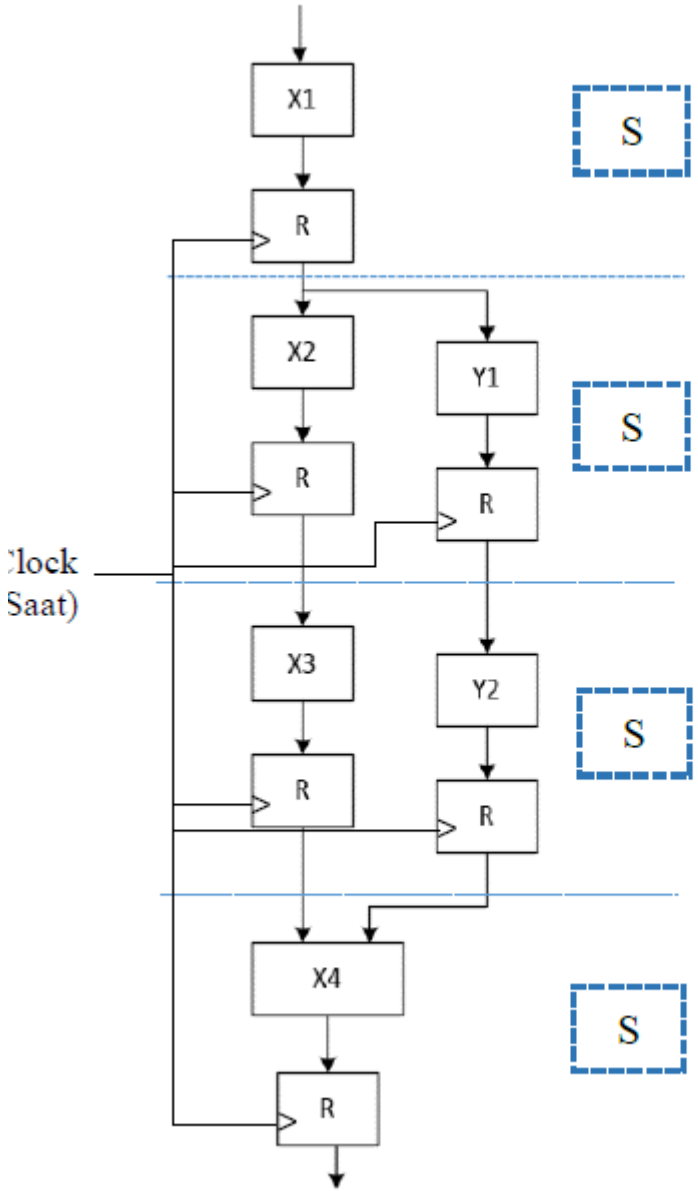
It's one solution to this question. You can find any other solution that gives the same amount of penalty.

Question 3

A task **T** that will be executed on integers, consists of six sub operations X_i (i=1,2,3,4) and Y_i (i=1,2). These sub operations are implemented using combinational digital circuits with the following propagation delays: X_1=25ns, X_2=15ns, X_3=20ns, X_4=40ns, Y_1=10ns, Y_2=10ns. The block diagram of the circuit is shown on the right. The sub operations should be executed in the given order. The sub operation Y_1 can be executed in parallel with X_2 or X_3. Similarly, the sub operation Y_2 can also be executed in parallel with X_2 or X_3.

To execute the task **T** faster on elements of an array, a pipeline with four stages (S_1, S_2, S_3, S_4) has been constructed as shown in the figure below. After each stage, a register with the delay of 5ns has been placed.





a) How long does it take to execute the task only on the first element using the pipeline (T_1)?

b) What is the duration of the one task without the pipeline (t_n)?

c) How many elements should the array at least have to gain a speedup with this pipeline?

d) What is the acquired speedup with this pipeline, when it executes a task T on an array having infinite elements?

e) Provide an alternative pipeline design that executes task **T** by considering following conditions:

- Do not decrease the speedup of the pipeline on an array having infinite elements.
- Use as minimum number of registers as possible.
- Use the units in the original circuit (X_i (i=1,2,3,4) and Y_i (i=1,2)).

f) How long does it take to execute the task only on the first element using the pipeline designed in (e) (T_1)?

g) When executing task T on an array having infinite elements, what is the acquired speedup with the pipeline designed in (e))?

Solution 3

a) The pipeline has 4 segments (k = 4) and the delay of the slowest segment (t_p) is:

$$S1 = X1 + R = 25 + 5 = 30 \text{ ns}$$
$$S2 = X2 + R = 15 + 5 = 20 \text{ ns}$$
$$S3 = X3 + R = 20 + 5 = 25 \text{ ns}$$
$$S4 = X4 + R = 40 + 5 = 45 \text{ ns}$$
$$\rightarrow \quad t_p = \max(S_i) = 45\ ns$$
$$T_1 = k \,.\, t_p = 4 \,.\, 45 = \mathbf{180\ ns}$$

b) Duration of the one task without the pipeline (t_n) is calculated by considering the longest path (without the registers).

$$t_n = 25 + 15 + 20 + 40 = \mathbf{100\ ns}$$

Solution 3

c) To gain a speedup with the pipeline, execution time without the pipeline should be bigger than the execution time with the pipeline.

$$n \,.\, \mathrm{t}_n > (k + n - 1).\, \mathrm{t}_p$$
$$n \,.\, 100 > (4 + n - 1).\, 45$$
$$n \,.\, 100 > 45.n + 135$$
$$55.n > 135$$
$$n > 2.4545 \ldots$$

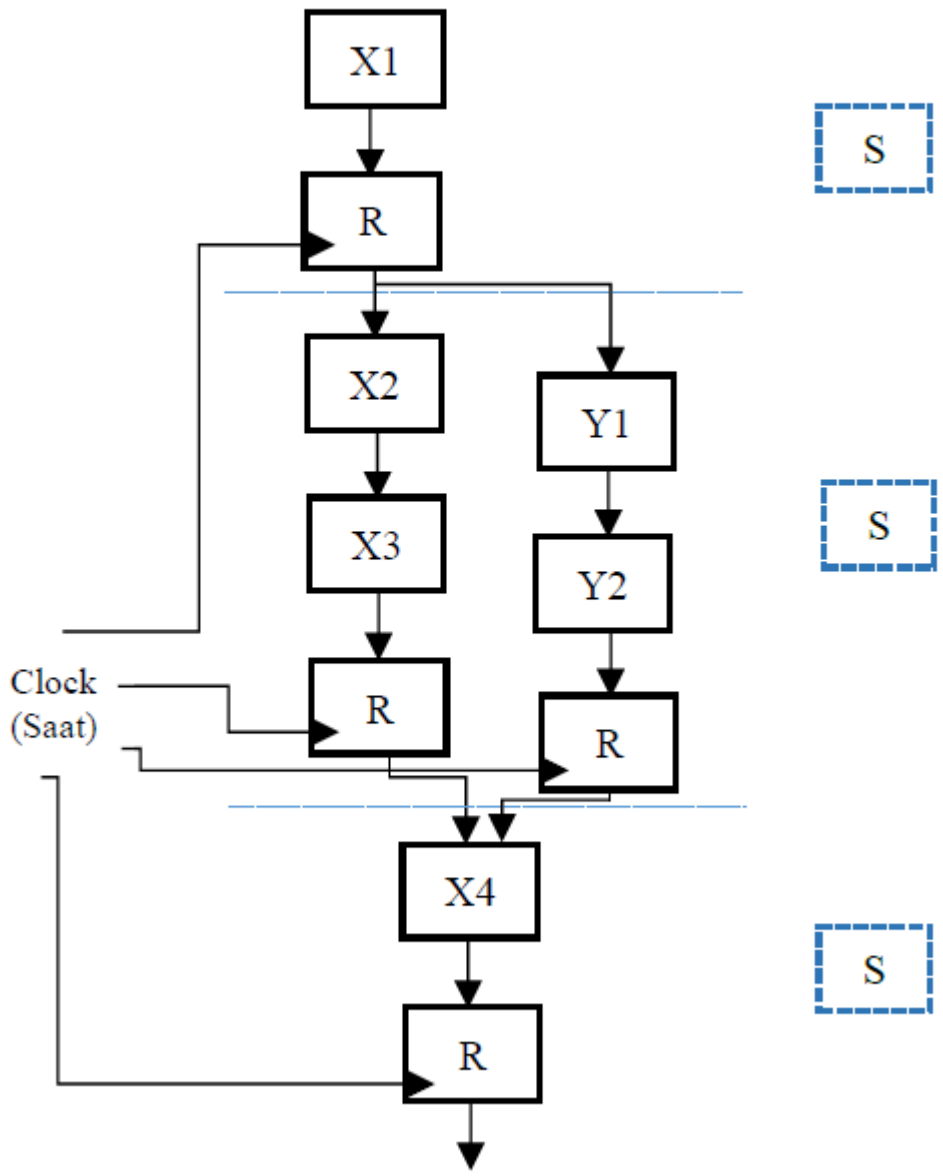
The array should have at least **3** elements to gain a speedup.

d)

$$\lim_{n \to \infty} S = \frac{\mathrm{t}_n}{\mathrm{t}_p} = \frac{100}{45} = \mathbf{2.222} \ldots$$

Solution 3

e) One possible pipeline design:



Solution 3

f) The pipeline has 3 segments (k = 3) and the delay of the slowest segment (t_p) is:

$$S1 = X1 + R = 25 + 5 = 30 \text{ ns}$$
$$S2 = X2 + X3 + R = 15 + 20 + 5 = 40 \text{ ns}$$
$$S3 = X4 + R = 40 + 5 = 45 \text{ ns}$$
$$\rightarrow \quad t_p = \max(S_i) = 45\ ns$$
$$T_1 = k \,.\, t_p = 3 \,.\, 45 = \mathbf{135\ ns}$$

g)

$$\lim_{n\to\infty} S = \frac{t_n}{t_p} = \frac{100}{45} = \mathbf{2.222\ldots}$$

Question 4

You will design a pipeline that will execute the operation $[2(-A_i)]^3$ where A is an array which consists of 8-bit signed numbers expressed by two's complementary method. (For simplicity, assume that $2A_i$ can fit in 8 bits). You are allowed to use only the components, which are given below with their timing attributes. You may use more than one of each unit if necessary.

- Memory, access time: 45 ns
- NOT gate, propagation delay: 10 ns
- Adder, propagation delay: 15 ns
- Shifter (combinatorial), propagation delay: 10 ns
- Multiplication circuit propagation delay: 45 ns
- Register, delay: 5 ns

a) Design and draw the optimum pipeline structure in terms of primarily speedup and secondarily implementation cost, also consider the waiting time for the first result.

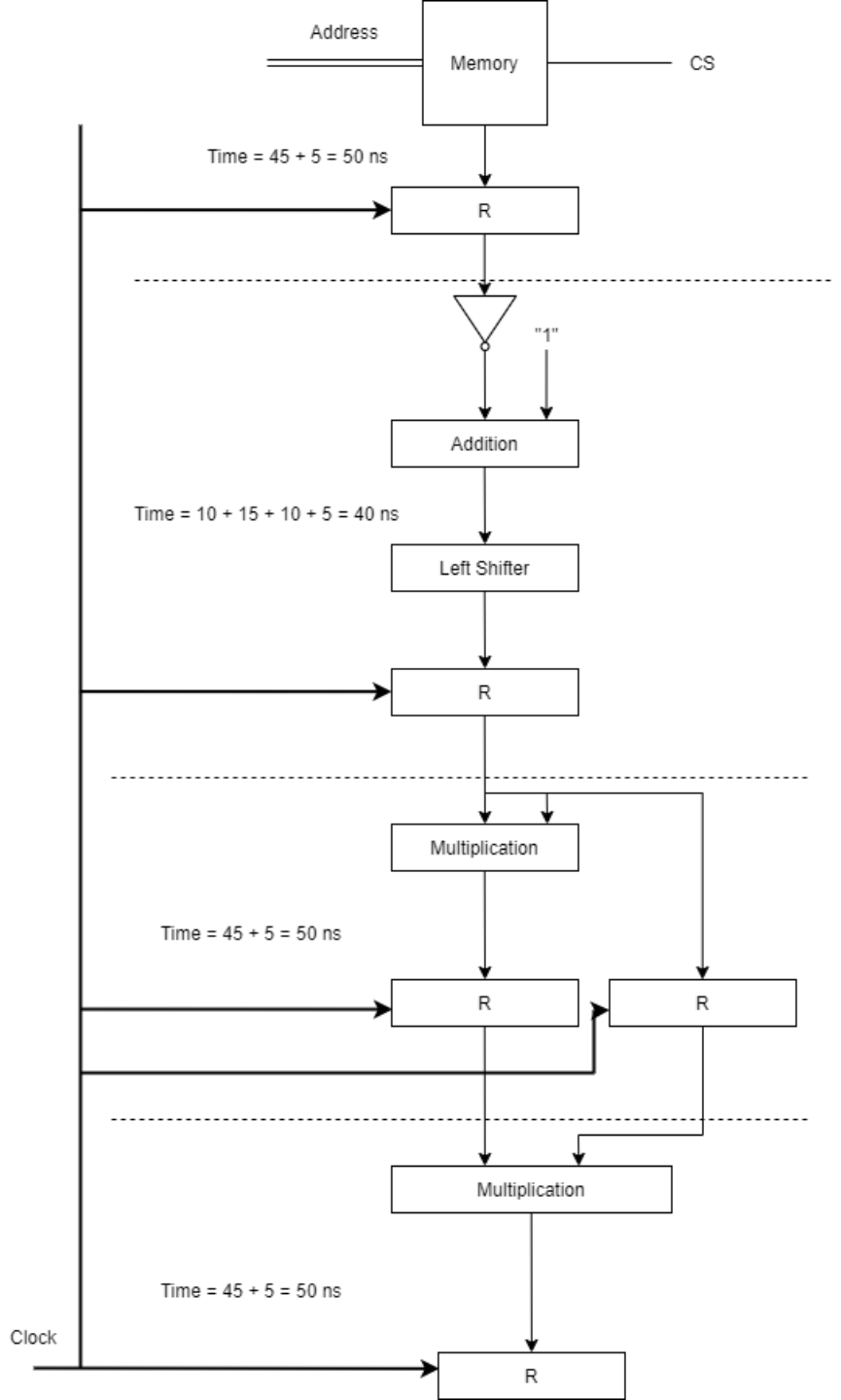
b) For the given propagation delay and access time information, calculate the speedup for an array of 8 signed numbers. Completion time without pipelining should be estimated as total latency of the combinatorial logic circuits in the longest way (without registers).

c) What is the theoritical speed up of this operation? Hint: consider the case that the number of array elements approaches infinity.

Solution 4

b) Required time for the slowest segment is 50 ns. Thus, the clock cycle should be the same, 50 ns.
Without pipeline -> **Tn = 45+10+15+10+45+45 = 170 ns**
Execution for 8 numbers with pipeline -> 4x50 + 7x50 = 550 ns
Speedup = $\frac{8x170}{550} = 2.47$

c) Theoretical Speedup = k = 4



Question 5

- A CPU that has an instruction pipeline with branch prediction mechanisms, runs the given piece of code below.

```
                Counter ← 10
LOOP:           ------              ; Any instruction
                Counter MOD 3;      ; Modulo 3 operation (Remainder of Counter/3)
                BNZ   L1:           ; Branch if NOT zero (if Counter is NOT divisible by 3)
                ------              ; Any instruction
L1:             ------              ; Any instruction
                Counter ← Counter - 1
                BNZ   LOOP          ; Branch if not zero
                ------              ; Instruction after the loop
```

- Fill in the tables below for both branch instructions that show the decision of the given prediction method, whether the branch is really taken or not, and if the prediction is correct or not for each run of the instructions. The first columns of the tables (for the first runs of the branch instructions) have been already filled.
- Assume that the branch history table includes the branch target address at the beginning.

Solution 5

a. Dynamic prediction with one bit, initial decision is to take the branch

i) BNZ L1

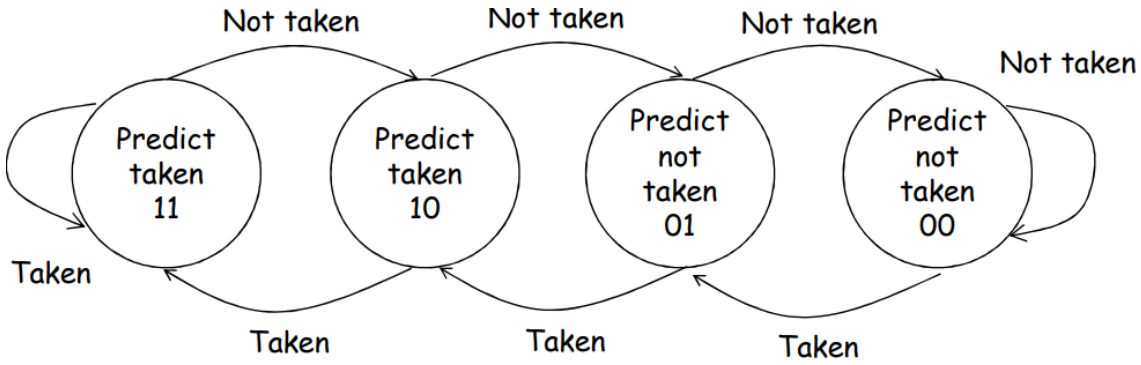
#run	1	2	3	4	5	6	7	8	9	10
Counter	10	9	8	7	6	5	4	3	2	1
Prediction (**T**aken or **N**ot)	T	T	N	T	T	N	T	T	N	T
Really **T**aken or **N**ot	T	N	T	T	N	T	T	N	T	T
Correct or **F**alse	C	F	F	C	F	F	C	F	F	C

```
        Counter ← 10
LOOP:   ------
        Counter MOD 3;
        BNZ  L1:
        ------
        ------
L1:     ------
        Counter ← Counter - 1
        BNZ  LOOP
        ------
```

ii) BNZ LOOP

#run	1	2	3	4	5	6	7	8	9	10
Counter	9	8	7	6	5	4	3	2	1	0
Prediction (**T**aken or **N**ot)	T	T	T	T	T	T	T	T	T	T
Really **T**aken or **N**ot	T	T	T	T	T	T	T	T	T	N
Correct or **F**alse	C	C	C	C	C	C	C	C	C	F

Solution 5



b. Dynamic prediction with two bits (saturating counter), initial decision is to take the branch (11)

i) BNZ L1

#run	1	2	3	4	5	6	7	8	9	10
Counter	10	9	8	7	6	5	4	3	2	1
State	11	11	10	11	11	10	11	11	10	11
Prediction (**T**aken or **N**ot)	T	T	T	T	T	T	T	T	T	T
Really **T**aken or **N**ot	T	N	T	T	N	T	T	N	T	T
Correct or **F**alse	C	F	C	C	F	C	C	F	C	C

```
        Counter ← 10
LOOP:   ------
        Counter MOD 3;
        BNZ   L1:
        ------
L1:     ------
        Counter ← Counter - 1
        BNZ   LOOP
        ------
```

ii) BNZ LOOP

#run	1	2	3	4	5	6	7	8	9	10
Counter	9	8	7	6	5	4	3	2	1	0
State	11	11	11	11	11	11	11	11	11	11
Prediction (**T**aken or **N**ot)	T	T	T	T	T	T	T	T	T	T
Really **T**aken or **N**ot	T	T	T	T	T	T	T	T	T	N
Correct or **F**alse	C	C	C	C	C	C	C	C	C	F