Computer Architecture Recitation 1

Prepared by Kadir OZLEM

(kadir.ozlem@itu.edu.tr)

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 $\mathbf{t}_n = 90$ns.

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 $\mathbf{P_{new}}$ 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 $\mathbf{T}_{new1} = 120$ 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* $\boldsymbol{t_n}$ *= 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- Branch Prediction

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 2

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

LOOP:
Counter ← 10

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 2

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

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

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

The End