BLG322E Computer Architecture
Assignment 1 Report

Mustafa Can Çalışkan
150200097

March 16, 2025

Contents

1 Solutions

1.1 How long does it take to execute the task only on the first element of array using the pipeline P_A?

To determine the time required to execute the task on the first element using pipeline P_A (denoted as T_1), we need to analyze the pipeline structure and calculate the total latency.

1.1.1 Processing Time for Each Stage

The processing time of each stage is determined by the longest suboperation within that stage:

1. S1: X1 = 5 ns
2. S2: $\max(X2, Y1) = \max(10 \text{ ns}, 30 \text{ ns}) = 30 \text{ ns}$
3. S3: $\max(X3, Y2, Z1) = \max(30 \text{ ns}, 20 \text{ ns}, 35 \text{ ns}) = 35 \text{ ns}$
4. S4: $\max(X4, Z2) = \max(40 \text{ ns}, 40 \text{ ns}) = 40 \text{ ns}$
5. S5: X5 = 15 ns

1.1.2 Register Delay

Between each stage, there is a register delay of 5ns.

1.1.3 Clock Period Calculation

The clock period of the pipeline is determined by:

$$\text{Clock period} = \text{Longest stage processing time} + \text{Register delay}$$

The longest stage is S4 with 40 ns, so:

$$T_{clk} = 40 \text{ ns} + 5 \text{ ns} = 45 \text{ ns} \tag{1}$$

1.1.4 First Element Execution Time

For the first element to complete processing, it must pass through all 5 stages of the pipeline. In an N-stage pipeline, the latency for the first result is given by:

$$\text{Latency for 1st result} = N \times T_{clk} \tag{2}$$

Where:

- N = 5 (number of stages)
- $T_{clk} = 45$ ns (clock period)

1.1.5 Conclusion

Therefore:

$$T_1 = 5 \times 45 \text{ ns} = 225 \text{ ns} \tag{3}$$

1.2 What is the duration of execution of the task T for once without the pipeline?

To determine the duration of execution of task T for once without the pipeline (denoted as t_1), we need to analyze the data dependencies between suboperations and find the critical path through the circuit.

1.2.1 Suboperation Execution Times

The task consists of nine suboperations with the following propagation delays:

- X1 = 5 ns
- X2 = 10 ns
- X3 = 30 ns
- X4 = 40 ns
- X5 = 15 ns
- Y1 = 30 ns
- Y2 = 20 ns
- Z1 = 35 ns
- Z2 = 40 ns

1.2.2 Data Dependencies

According to the dependency diagram, the following dependencies exist:

1. X1 $\rightarrow$ X2 and X1 $\rightarrow$ Y1
2. X2 $\rightarrow$ X3 and X2 $\rightarrow$ Z1
3. X3 $\rightarrow$ X4
4. X4 $\rightarrow$ X5
5. Y1 $\rightarrow$ Y2
6. Y2 $\rightarrow$ X4
7. Z1 $\rightarrow$ Z2
8. Z2 $\rightarrow$ X5

1.2.3 Critical Path Analysis

To find the total execution time without pipelining, we need to identify all possible paths from the starting operation (X1) to the final output (X5) and determine which path takes the longest time.

1.2.4 Identifying Possible Paths

Based on the dependencies, there are three main paths from X1 to X5:

1. Path 1: X1 → X2 → X3 → X4 → X5
   - Duration: 5 + 10 + 30 + 40 + 15 = 100 ns
2. Path 2: X1 → Y1 → Y2 → X4 → X5
   - Duration: 5 + 30 + 20 + 40 + 15 = 110 ns
3. Path 3: X1 → X2 → Z1 → Z2 → X5
   - Duration: 5 + 10 + 35 + 40 + 15 = 105 ns

1.2.5 Conclusion

The critical path is the longest path through the circuit, which in this case is Path 2 with a duration of 110 ns. Therefore, the duration of execution of task T for once without the pipeline is 110 ns.

1.3 What is the minimum number of elements the array should have to achieve any speedup with this pipeline?

To determine the minimum number of elements the array should have to achieve any speedup with pipeline P_A, we need to compare the execution times with and without pipelining for various array sizes.

1.3.1 Known Parameters

From previous calculations, we have:

- Non-pipelined execution time for one element (t_1) = 110 ns
- Pipeline latency for the first element (T_1) = 225 ns
- Pipeline clock period (T_{clk}) = 45 ns

1.3.2 Execution Time Formulas

- Without pipeline: $T_{non-pipe}(N) = N \times t_1 = 110N$ ns
- With pipeline PA: $T_{pipe}(N) = T_1 + (N-1) \times T_{clk} = 225 + 45(N-1)$ ns
- Simplified: $T_{pipe}(N) = 225 + 45N - 45 = 180 + 45N$ ns

1.3.3 Finding the Break-Even Point

For the pipeline to achieve any speedup:

$$T_{pipe}(N) < T_{non-pipe}(N) \tag{4}$$
$$180 + 45N < 110N \tag{5}$$
$$180 < 110N - 45N \tag{6}$$
$$180 < 65N \tag{7}$$
$$\frac{180}{65} < N \tag{8}$$
$$2.77 < N \tag{9}$$

Since N must be an integer (number of array elements), the smallest value of N for which pipelining is faster is:

$$N = 3 \tag{10}$$

1.3.4 Verification

Let's verify by calculating the execution times for $N = 2$ and $N = 3$:
For $N = 2$:

- Without pipeline: $T_{non-pipe}(2) = 110 \times 2 = 220$ ns
- With pipeline: $T_{pipe}(2) = 180 + 45 \times 2 = 270$ ns
- Result: Pipeline is slower (270 ns > 220 ns)

For $N = 3$:

- Without pipeline: $T_{non-pipe}(3) = 110 \times 3 = 330$ ns
- With pipeline: $T_{pipe}(3) = 180 + 45 \times 3 = 315$ ns
- Result: Pipeline is faster (315 ns < 330 ns)

1.3.5 Conclusion

The minimum number of elements the array should have to achieve any speed-up with this pipeline is 3.

1.4 Calculate the highest possible speedup pipeline P_A can achieve when it executes a task T on an array with an infinite number of elements.

To calculate the highest possible speedup that pipeline PA can achieve when executing task T on an array with an infinite number of elements, we need to determine the limit of the ratio between non-pipelined and pipelined execution times as the number of elements approaches infinity.

1.4.1 Known Parameters

From previous calculations:

- Non-pipelined execution time per element (t_1) = 110 ns
- Pipeline clock period (T_{clk}) = 45 ns (determined by the slowest stage + register delay)

1.4.2 Execution Time Formulas

For N elements:

- Without pipeline: $T_{non-pipe}(N) = N \times t_1 = 110N$ ns
- With pipeline PA: $T_{pipe}(N) = T_1 + (N-1) \times T_{clk}$
- where $T_1 = 225$ ns (first element latency) and $T_{clk} = 45$ ns

1.4.3 Calculating Maximum Speedup

As N approaches infinity, the speedup approaches:

$$\begin{aligned}
\lim_{N\to\infty} \text{Speedup} &= \lim_{N\to\infty} \frac{T_{non-pipe}(N)}{T_{pipe}(N)} && (11)\\
&= \lim_{N\to\infty} \frac{110N}{225 + 45(N-1)} && (12)\\
&= \lim_{N\to\infty} \frac{110N}{180 + 45N} && (13)\\
& && (14)
\end{aligned}$$

For very large N, we can simplify this expression:

$$\begin{aligned}
\lim_{N\to\infty} \frac{110N}{180 + 45N} &= \lim_{N\to\infty} \frac{110}{\frac{180}{N} + 45} && (15)\\
&= \frac{110}{0 + 45} && (16)\\
&= \frac{110}{45} && (17)\\
&= 2.44\overline{4} && (18)
\end{aligned}$$

1.4.4 Verification

This maximum speedup occurs because:

- In non-pipelined execution, each element requires 110 ns
- In pipelined execution with an infinite array, after the initial latency, we complete one element every 45 ns (the pipeline clock period)
- The ratio $\frac{110}{45} \approx 2.44$ represents how many times faster we can process elements once the pipeline is fully operational

1.4.5 Conclusion

The highest possible speedup pipeline PA can achieve when executing task T on an array with an infinite number of elements is:

$$\text{Maximum Speedup} = \frac{110 \text{ ns}}{45 \text{ ns}} = 2.44\overline{4} \approx 2.44 \tag{19}$$

1.5 Alternative Pipeline P_B

According to the provided guidelines, the designed alternative pipeline P_B is shown below. In this pipeline, the number of stages has been reduced by incorporating the X_1 suboperation into the stage below it, resulting in savings in the number of registers.

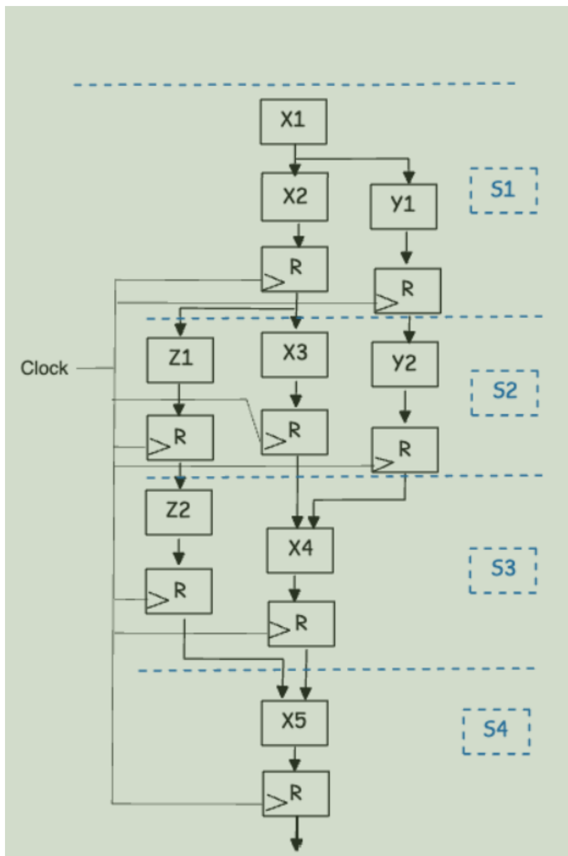

Figure 1: Design of alternative pipeline P_B with reduced number of stages and registers

1.6 How long does it take to execute the task only on the first element using pipeline P_B?

To determine how long it takes to execute the task on the first element using pipeline P_B, we need to analyze the pipeline structure and calculate the latency for one element to pass through all stages.

1.6.1 Pipeline P_B Structure

From the provided diagram, pipeline P_B consists of 4 stages:

- Stage 1 (S1): X1 $\rightarrow$ X2 and X1 $\rightarrow$ Y1 (in parallel)
- Stage 2 (S2): X3, Y2, and Z1 (in parallel)
- Stage 3 (S3): X4 and Z2 (in parallel)
- Stage 4 (S4): X5

1.6.2 Execution Time Calculation

1.6.3 Stage Processing Times

The processing time of each stage is determined by its longest suboperation:

- Stage 1: $\max(5 + 10, 5 + 30) = \max(15, 35) = 35$ ns
- Stage 2: $\max(30, 20, 35) = 35$ ns
- Stage 3: $\max(40, 40) = 40$ ns
- Stage 4: 15 ns

1.6.4 Clock Period

The clock period of the pipeline is determined by the slowest stage plus the register delay:

$$T_{clk} = 40 + 5 = 45 \text{ ns} \tag{20}$$

1.6.5 First Element Latency

For the first element to complete processing, it must pass through all 4 stages of the pipeline. In an N-stage pipeline, the latency for the first result is:

$$T_1 = N \times T_{clk} = 4 \times 45 = 180 \text{ ns} \tag{21}$$

1.6.6 Conclusion

The time to execute the task only on the first element using pipeline P_B (T_1) = 180 ns.

1.7 When executing task T on an array with an infinite number of elements, what speedup does the pipeline P_B?

To calculate the speedup that pipeline P_B achieves when executing task T on an array with an infinite number of elements, we need to determine the ratio between non-pipelined execution time and pipelined execution time as the number of elements approaches infinity.

1.7.1 Known Parameters

From the given diagram and previous calculations:

- Non-pipelined execution time per element (t_1) = 110 ns
- Pipeline P_B has 4 stages (S1-S4)
- The slowest stage in P_B is Stage 3 with 40 ns
- Register delay = 5 ns
- Pipeline clock period (T_{clk}) = 40 + 5 = 45 ns

1.7.2 Execution Time Formulas

For N elements:

- Without pipeline: $T_{non-pipe}(N) = N \times t_1 = 110N$ ns
- With pipeline P_B: $T_{pipe}(N) = T_1 + (N-1) \times T_{clk}$
- where $T_1 = 180$ ns (first element latency) and $T_{clk} = 45$ ns

1.7.3 Calculating Speedup for Infinite Elements

As N approaches infinity, the speedup approaches:

$$\begin{aligned}
\lim_{N\to\infty} \text{Speedup} &= \lim_{N\to\infty} \frac{T_{non-pipe}(N)}{T_{pipe}(N)} && (22)\\
&= \lim_{N\to\infty} \frac{110N}{180 + 45(N-1)} && (23)\\
&= \lim_{N\to\infty} \frac{110N}{135 + 45N} && (24)\\
& && (25)
\end{aligned}$$

For very large N, we can simplify this expression:

$$\lim_{N\to\infty} \frac{110N}{135+45N} = \lim_{N\to\infty} \frac{110}{\frac{135}{N}+45} \tag{26}$$

$$= \frac{110}{0+45} \tag{27}$$

$$= \frac{110}{45} \tag{28}$$

$$= 2.44\overline{4} \tag{29}$$

1.7.4 Comparison with Pipeline P_A

Both pipelines P_A and P_B have the same clock period of 45 ns, which is determined by the slowest stage (40 ns) plus the register delay (5 ns). Therefore, they achieve the same maximum throughput of one element every 45 ns once the pipeline is full.

1.7.5 Conclusion

The speedup pipeline P_B achieves when executing task T on an array with an infinite number of elements is:

$$\text{Speedup}_{P_B} = \frac{110 \text{ ns}}{45 \text{ ns}} = 2.44\overline{4} \approx 2.44 \tag{30}$$

This is exactly the same speedup as pipeline P_A. The advantage of P_B is that it achieves this same speedup while using fewer pipeline registers and stages.