## Seminar 2

## 1. Instruction Pipelining <br> 2. Superscalars <br> 3. Parallel Architectures

## Problem 1

A nonpipelined processor has a clock rate of 2.5 GHz . An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, the clock rate of the new processor has to be reduced to 2 GHz .

What is the speedup achieved for a sequence of 100 instructions?

## Problem 1

A nonpipelined processor has a clock rate of 2.5 GHz . An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, the clock rate of the new processor has to be reduced to 2 GHz .

What is the speedup achieved for a sequence of 100 instructions?

## Remember from Lecture 4-5:

ㅁ $\tau$ : duration of one cycle

- $n$ : number of instructions to execute
$\square k$ : number of pipeline stages
- $T_{k, n}$ : total time to execute $n$ instructions on a pipeline with $k$ stages
$\square S_{k, n}$ : (theoretical) speedup produced by a pipeline with $k$ stages when executing $\boldsymbol{n}$ instructions

$$
T_{k, n}=[k+(n-1)] \times \tau
$$

On a non-pipelined processor each instruction takes $\boldsymbol{k} \times \tau$, and $n$ instructions take $T_{n}=n \times k \times \tau$

## Problem 1

A nonpipelined processor has a clock rate of 2.5 GHz . An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, the clock rate of the new processor has to be reduced to 2 GHz .

What is the speedup achieved for a sequence of 100 instructions?

## Remember from Lecture 4-5:

ㅁ $\tau$ : duration of one cycle

- n: number of instructions to execute
- $k$ : number of pipeline stages

व $T_{k, n}$ : total time to execute $n$ instructions on a pipeline with $k$ stages

- $S_{k, n}$ : (theoretical) speedup produced by a pipeline with $k$ stages when executing $\boldsymbol{n}$ instructions

$$
T_{k, n}=[k+(n-1)] \times \tau
$$

On a non-pipelined processor each instruction takes $\boldsymbol{k} \times \tau$, and $\boldsymbol{n}$ instructions take $\boldsymbol{T}_{\boldsymbol{n}}=\boldsymbol{n} \times \boldsymbol{k} \times \tau$

$$
S_{k, n}=\frac{T_{n}}{T_{k, n}}=\frac{n \times k \times \tau}{[k+(n-1)] \times \tau}=\frac{n \times k}{k+(n-1)}
$$

## Problem 1

A nonpipelined processor has a clock rate of 2.5 GHz . An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, the clock rate of the new processor has to be reduced to 2 GHz .

What is the speedup achieved for a sequence of 100 instructions?
Solution

$$
S_{5,100}=\frac{100 \times 5}{5+(100-1)}=4,8
$$

We would have a speedup of 4.8 if the pipelined processor would work at the same clock rate as the initial one!

But the clock rate of the pipelined processor is reduced by a factor of 2/2.5 = 0.8


$$
S=4.8 \times 0.8=3.8
$$

## Problem 2

Consider the following assembly language program:

| 1: Move R3, R7 | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ |
| :--- | :--- |
| 2: Load R8, (R3) | $\mathrm{R} 8 \leftarrow(\mathrm{R} 3)$ |
| 3: Add R3, R3,4 | $\mathrm{R} 3 \leftarrow \mathrm{R} 3+4$ |
| 4: Load R9, (R3) | $\mathrm{R} 9 \leftarrow(\mathrm{R} 3)$ |
| 5: BLE R8, R9, L3 | Branch if $\mathrm{R} 9>\mathrm{R} 8$ |

Show the dependencies.

## Problem 2

Consider the following assembly language program:

| ( 1: Move R3, R7 | R3 $\leftarrow \mathrm{R} 7$ |  |
| ---: | :--- | :--- |
| 2: Load R8, (R3) | $\mathrm{R} 8 \leftarrow(\mathrm{R} 3)$ |  |
| 3: Add R3, R3, 4 | $\mathrm{R} 3 \leftarrow \mathrm{R} 3+4$ |  |
| 4: Load R9, (R3) | $\mathrm{R} 9 \leftarrow(\mathrm{R} 3)$ |  |
| 5: BLE R8, R9, L3 |  | Branch if $\mathrm{R} 9>\mathrm{R} 8$ |

Show the dependencies.

## Problem 2

Consider the following assembly language program:
$\left(\begin{array}{ll}\text { 1: } \text { Move R3, R7 } & \mathrm{R} 3 \leftarrow \mathrm{R} 7 \\ \text { 2: } \text { Load R8, (R3) } & \mathrm{R} 8 \leftarrow(\mathrm{R} 3) \\ \text { 3: Add R3, R3, 4 } & \mathrm{R} 3 \leftarrow \mathrm{R} 3+4 \\ \text { 4: } \text { Load R9, (R3) } & \mathrm{R} 9 \leftarrow(\mathrm{R} 3) \\ \text { 5: } \text { BLE R8, R9, L3 } & \text { Branch if } \mathrm{R} 9>\mathrm{R} 8\end{array}\right.$

Show the dependencies.

## Problem 2

Consider the following assembly language program:
$\left(\begin{array}{ll}\text { 1: Move R3, R7 } & R 3 \leftarrow R 7 \\ \text { 2: Load R8, (R3) } & R 8 \leftarrow(R 3) \\ \text { ( 3: Add R3, R3, 4 } & R 3 \leftarrow R 3+4 \\ \text { 4: Load R9, (R3) } & R 9 \leftarrow(R 3) \\ \text { 5: } \text { BLE R8, R9, L3 } & \text { Branch if } R 9>R 8\end{array}\right.$

Show the dependencies.

## Problem 2

Consider the following assembly language program:
$\left(\begin{array}{ll}\text { 1: } \text { Move R3, R7 } & R 3 \leftarrow R 7 \\ \text { 2: Load R8, (R3) } & R 8 \leftarrow(R 3) \\ \text { ( 3: Add R3, R3, 4 } & R 3 \leftarrow R 3+4 \\ \text { 4: Load R9, (R3) } & R 9 \leftarrow(R 3) \\ \text { 5: } \text { BLE R8, R9, L3 } & \text { Branch if } R 9>R 8\end{array}\right.$

Show the dependencies.

## Problem 2

Consider the following assembly language program:
$\left(\begin{array}{ll}\text { 1: } \text { Move R3, R7 } & R 3 \leftarrow R 7 \\ \text { 2: Load R8, (R3) } & R 8 \leftarrow(R 3) \\ \text { 3: Add R3, R3, 4 } & R 3 \leftarrow R 3+4 \\ \text { 4: Load R9, (R3) } & R 9 \leftarrow(R 3) \\ \text { 5: } \text { BLE R8, R9, L3 } & \text { Branch if } R 9>R 8\end{array}\right.$

Show the dependencies.

## Problem 2

Consider the following assembly language program:
$\left(\begin{array}{ll}\text { 1: Move R3, R7 } & R 3 \leftarrow R 7 \\ \text { 2: Load R8, (R3) } & R 8 \leftarrow(R 3) \\ \text { 3: Add R3, R3, 4 } & R 3 \leftarrow R 3+4 \\ \text { 4: Load R9, (R3) } & R 9 \leftarrow(R 3) \\ \text { 5: BLE R8, R9, L3 } & \text { Branch if } R 9>R 8\end{array}\right.$

Show the dependencies.

True data dependency (RAW): 1-2, 1-3, 2-5, 3-4, 4-5
Output dependency (WAW): 1-3
Antidependency (WAR): 2-3

## Problem 3

a. Identify the dependencies in the following code:

1: $\quad \mathrm{R} 1 \leftarrow 100$
2: $\quad \mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$
3: $\quad \mathrm{R} 7 \leftarrow \mathrm{R} 5+1$
4: $\quad \mathrm{R} 1 \leftarrow \mathrm{R} 2+\mathrm{R} 4$
5: $\quad \mathrm{R} 5 \leftarrow 0$
6: $\quad \mathbf{R} 2 \leftarrow R 4-25$
7: $\quad R 3 \leftarrow R 7-2$
8: $\quad \mathrm{R} 4 \leftarrow \mathrm{R} 1+\mathrm{R} 3$
9: $\mathrm{R} 10 \leftarrow 0$
10: $\mathbf{R 1} \leftarrow \mathbf{R} 1+30$
b. Rename the registers in the above sequence to prevent, where possible, dependency problems.
c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with:
c1) in order execution;
c2) out of order execution before renaming;
c3) out of order execution after renaming.

## Problem 3

a. Identify the dependencies in the following code:

```
1: }\textrm{R}1\leftarrow10
2: R5 \leftarrowR1 + R2
\3: R7 \leftarrowR5 + 1
4: R1 \leftarrowR2 + R4
5: R5}\leftarrow
6: R2}\leftarrowR4-2
7: R3\leftarrowR7-2
8: R4}\leftarrow\textrm{R}1+\textrm{R}
9: R10}\leftarrow
10: R1 \leftarrow R1 + 30
```


## Problem 3

a. Identify the dependencies in the following code:

```
1: }\textrm{R}1\leftarrow10
2: R5 \leftarrowR1 + R2
13: R7 }\leftarrowR55+
4: R1 \leftarrowR2 + R4
5: R5}\leftarrow
6: R2 \leftarrowR4-25
17: R3\leftarrowR7 - 2
8: R4}\leftarrowR1 + R
9: R10}\leftarrow
10: R1 \leftarrow R1 + 30
```


## Problem 3

a. Identify the dependencies in the following code:

$$
\begin{aligned}
\text { 1: } & \mathrm{R} 1 \leftarrow 100 \\
\text { 2: } & \mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2 \\
\text { 3: } & \mathrm{R} 7 \leftarrow \mathrm{R} 5+1 \\
\text { 4: } & \mathrm{R} 1 \leftarrow \mathrm{R} 2+\mathrm{R} 4 \\
5: & \mathrm{R} 5 \leftarrow 0 \\
6: & \mathrm{R} 2 \leftarrow \mathrm{R} 4-25 \\
\text { 7: } & \mathrm{R} 3 \leftarrow \mathrm{R} 7-2 \\
8: & \mathrm{R} 4 \leftarrow \mathrm{R} 1+\mathrm{R} 3 \\
\text { 9: } & \mathrm{R} 10 \leftarrow 0 \\
10: & \mathrm{R} 1 \leftarrow \mathrm{R} 1+30
\end{aligned}
$$

## Problem 3

a. Identify the dependencies in the following code:

| 1 | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
| 2: | $\mathbf{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
| 3: | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
| 4: | $\mathbf{R} 1 \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4$ |
| $5:$ | $\mathrm{R} 5 \leftarrow 0$ |
| 6: | $\mathbf{R} 2 \leftarrow$ R $4-25$ |
| 7: | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
| 8: | $\mathbf{R} 4 \leftarrow \mathbf{R} 1+\mathrm{R} 3$ |
| 9: | $\mathrm{R} 10 \leftarrow 0$ |
| 10 | $\mathbf{R 1} \leftarrow \mathbf{R} 1+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.

## Problem 3

a. Identify the dependencies in the following code:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

b. Rename registers to prevent output - and antidependencies:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

b. Rename registers to prevent output - and antidependencies:


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $1 / 4,1$-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
| 12: | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
| 3 : | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
| 4: | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R 4}$ |
| 5: | $\rightarrow \mathrm{R} 5 \leftarrow 0$ |
| 6 : | R2 $\leftarrow$ R $4-25$ |
| 7: | $\mathbf{R} 3 \leftarrow \mathbf{R} 7$ - 2 |
| 8: | R4 $\leftarrow$ R11 + R3 |
| 9: | $\mathrm{R} 10 \leftarrow 0$ |
|  | $\mathbf{R} 1 \leftarrow \mathrm{R} 11+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1-10,2-5,4<10$.
Antidependency (WAR): 3-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8>10

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\begin{aligned} & \mathrm{R} 1 \leftarrow 100 \\ & \mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} \end{aligned}$ |
| :---: | :---: |
| 13: | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
| 4: | $\mathrm{R} 11 \leftarrow \mathrm{R} 2+\mathrm{R} 4$ |
| 5 : | R5 $\leftarrow 0$ |
| 6 | R2 $\leftarrow R 4-25$ |
| 7: | $\mathrm{R} 3 \leftarrow \mathrm{R} 7-2$ |
| 8. | $\mathrm{R} 4 \leftarrow \mathrm{R} 11+\mathrm{R} 3$ |
| 9 9: | $\mathrm{R} 10 \leftarrow 0$ |
|  | $\mathbf{R} 1 \leftarrow \mathrm{R} 11+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1-10,2-5,4<10$.
Antidependency (WAR): 3-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8>10

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
|  | $\mathbf{R} 5 \leftarrow \mathbf{R} 1$ |
|  | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R 4}$ |
|  | $\rightarrow \mathrm{R} 5 \leftarrow 0$ |
| 6: | R2 $\leftarrow$ R 4 - 25 |
| 7: | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
| 18. | R4 $\leftarrow$ R11 + R3 |
|  | $10 \leftarrow 0$ |
|  | $12 \leftarrow \mathrm{R} 11$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1>10,2-5,4<10$.
Antidependency (WAR): 3 4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8>10

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
| , | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R}$ |
|  | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathrm{R} 2+\mathrm{R} 4$ |
|  | $\checkmark R 5 \leftarrow 0$ |
| 6: | - $\mathrm{R} 2 \leftarrow \mathrm{R} 4-25$ |
| 7. | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
|  | R4 $\leftarrow$ R11 + R3 |
|  | $\mathbf{R 1 0} \leftarrow 0$ |
|  | $\mathrm{R} 12 \leftarrow \mathrm{R} 11$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1>10,2-5,4<10$.
Antidependency (WAR): $3>4,2><0,2-6,3-5,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

| 1: | $R 1 \leftarrow 100$ |
| ---: | :--- |
| 2: | $R 5 \leftarrow R 1+R 2$ |
| 3: | $R 7 \leftarrow R 5+1$ |
| 4: | $R 11 \leftarrow R 2+R 4$ |
| 5: | $R 2 \leftarrow 0$ |
| 6: | $R 2 \leftarrow R 4-25$ |
| $17:$ | $R 3 \leftarrow R 7-2$ |
| $18:$ | $R 4 \leftarrow R 11+R 3$ |
| $9:$ | $R 10 \leftarrow 0$ |
| $10:$ | $R 12 \leftarrow R 11+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1>10,2-5,4<10$.
Antidependency (WAR): $3>4,2><0,2-6,3-5,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
|  | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
|  | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4$ |
| $5:$ | $\checkmark$ R51 $\leftarrow 0$ |
| 6: | R2 $\leftarrow$ R $4-25$ |
| 7: | $\mathbf{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
| 18 | R4 $\leftarrow$ R11 + R3 |
|  | $\mathrm{R} 10 \leftarrow 0$ |
| 10. | $\mathbf{R 1 2} \leftarrow \mathbf{R 1 1}+$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2-6,3-5,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
|  | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
|  | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathrm{R} 4$ |
| 5: | $\mathrm{R} 51 \leftarrow 0$ |
| 6: | R2 $\leftarrow$ R4-25 |
| 7: | $\mathbf{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
| 18 | R4 $\leftarrow$ R11 + R3 |
|  | $\mathrm{R} 10 \leftarrow 0$ |
| 10 | $\mathrm{R} 12 \leftarrow \mathrm{R} 11+$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2-6,3>2,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

|  | $\leftarrow 100$ |
| :---: | :---: |
|  | $R 5 \leftarrow R 1+R 2$ |
|  | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R} 11 \leftarrow \mathbf{R} 2+\mathbf{R} 4$ |
| 5: | $\mathrm{R} 51 \leftarrow 0$ |
|  | $\mathbf{R 2} \leftarrow \mathbf{R} 4$ - 25 |
| 7: | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
|  | R4 $\leftarrow$ R11 + R3 |
|  | $\mathrm{R} 10 \leftarrow$ |
|  | $\mathrm{R} 12 \leftarrow \mathrm{R} 11+$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2-6,3>2,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

| 1: | $R 1 \leftarrow 100$ |
| ---: | :--- |
| 2: | $R 5 \leftarrow R 1+R 2$ |
| 3: | $R 7 \leftarrow R 5+1$ |
| 4: | $R 11 \leftarrow R 2+R 4$ |
| 5: | $R 51 \leftarrow 0$ |
| 6: | $R 21 \leftarrow R 4-25$ |
| 7: | $R 3 \leftarrow R 7-2$ |
| $18:$ | $R 4 \leftarrow R 11+R 3$ |
| $9:$ | $R 10 \leftarrow 0$ |
| $10:$ | $R 12 \leftarrow R 11+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2<6,3>2,4-6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

| $1:$ | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
| 2: | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
| $13:$ | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4$ |
| 5: | R51 $\leftarrow 0$ |
| 6: | $\mathbf{R 2 1} \leftarrow \mathbf{\leftarrow} \mathbf{R}$ - 25 |
|  | $\mathrm{R} 3 \leftarrow \mathrm{R} 7$ - 2 |
|  | R4 $\leftarrow$ R11 + R3 |
|  | $\mathrm{R} 10 \leftarrow 0$ |
| $10:$ | $\mathrm{R} 12 \leftarrow \mathrm{R} 11+$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2<6,3><, 4<6,4-8,6-8,8><0$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

| 1: | $\mathrm{R} 1 \leftarrow 100$ |
| :--- | :--- |
| 2: | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
| 3: | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
| 4: | $\mathrm{R} 11 \leftarrow \mathrm{R} 2+\mathrm{R} 4$ |
| 5: | $\mathrm{R} 51 \leftarrow 0$ |
| 6: | $\mathrm{R} 21 \leftarrow \mathrm{R} 4-25$ |
| $7:$ | $\mathrm{R} 3 \leftarrow \mathrm{R} 7-2$ |
| $1:$ | $\mathrm{R} 4 \leftarrow \mathrm{R} 11+\mathrm{R} 3$ |
| $9:$ | $\mathrm{R} 10 \leftarrow 0$ |
| $10:$ | $\mathrm{R} 12 \leftarrow \mathrm{R} 11+30$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4 \ll 0$.
Antidependency (WAR): $3>4,2><0,2<6,3><, 4<6,4-8,6-8,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

| 1 1: | $\mathrm{R} 1 \leftarrow 100$ |
| :---: | :---: |
| . | $\mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2$ |
| $13:$ | $\mathrm{R} 7 \leftarrow \mathrm{R} 5+1$ |
|  | $\mathbf{R 1 1} \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4$ |
| 5: | $\mathrm{R} 51 \leftarrow 0$ |
| 6: | $\mathbf{R 2 1} \leftarrow \mathbf{R} 4$ - 25 |
|  | $\mathbf{R} 3 \leftarrow \mathbf{R} 7$ - 2 |
|  | 'R41 $\leftarrow$ R11 |
|  | $\mathrm{R} 10 \leftarrow 0$ |
| 10: | $\mathbf{R 1 2} \leftarrow \mathrm{R} 11+$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): $><4,1><0,2>5,4<10$.
Antidependency (WAR): $3>4,2><0,2<6,3<2,4<6,4 \ll 6,6-8,8><0$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

```
1: \(\quad \mathrm{R} 1 \leftarrow 100\)
    \(\mathbf{R} 5 \leftarrow \mathbf{R} 1+\mathbf{R} \mathbf{2}\)
    \(R 7 \leftarrow R 5+1\)
    \(\mathbf{R} 11 \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4\)
    \(\mathrm{R} 51 \leftarrow 0\)
    R21 \(\leftarrow \mathbf{R} 4-25\)
    \(\mathbf{R} 3 \leftarrow \mathbf{R} 7-2\)
    \(\mathbf{R} 41 \leftarrow \mathbf{R} 11+\mathbf{R} 3\)
    \(\mathbf{R 1 0} \leftarrow \mathbf{0}\)
    \(\mathbf{R} 12 \leftarrow \mathbf{R 1 1}+30\)
```

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .
Output dependency (WAW): $><4,1><0,2>5,4<10$. Antidependency (WAR): $3>4,2><0,2<6,3><, 4<6,4<6,6<6,8>10$

## Problem 3

b. Rename registers to prevent output - and antidependencies:

```
1: \(\quad \mathrm{R} 1 \leftarrow 100\)
    \(\mathbf{R} 5 \leftarrow \mathbf{R} 1+\mathbf{R} \mathbf{2}\)
    \(R 7 \leftarrow R 5+1\)
    \(\mathbf{R} 11 \leftarrow \mathbf{R} \mathbf{2}+\mathbf{R} 4\)
    \(\mathrm{R} 51 \leftarrow 0\)
    R21 \(\leftarrow \mathbf{R} 4-25\)
    \(\mathbf{R} 3 \leftarrow \mathbf{R} 7-2\)
    \(\mathbf{R} 41 \leftarrow \mathbf{R} 11+\mathbf{R} 3\)
    \(\mathbf{R 1 0} \leftarrow \mathbf{0}\)
    \(\mathbf{R} 12 \leftarrow \mathbf{R 1 1}+30\)
```

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .
Output dependency (WAW): $><4,1><0,2>5,4<10$. Antidependency (WAR): $3>4,2><0,2<6,3><, 4<6,4<6,6<6,8>10$

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :--- | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 5 | 5 | 6 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |
| Cycle 5 | 7 |  |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |
| Cycle 5 | 7 |  |
| Cycle 6 | 8 | 9 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c1) in order execution;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 |  |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |
| Cycle 5 | 7 |  |
| Cycle 6 | 8 | 9 |
| Cycle 7 | 10 |  |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 9 |
| Cycle 2 | 2 |  |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 9 |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 9 |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 9 |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |
| Cycle 5 | 7 |  |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c2) out-of-order order execution, without renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 9 |
| Cycle 2 | 2 |  |
| Cycle 3 | 3 | 4 |
| Cycle 4 | 5 | 6 |
| Cycle 5 | 7 |  |
| Cycle 6 | 8 | $10!$ |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.
Output dependency (WAW): 1-4, 1-10, 2-5, 4-10.
Antidependency (WAR): 2-4, 2-10, 2-6, 3-5, 4-6, 4-8, 6-8, 8-10

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c3) out-of-order order execution, with renaming;


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.

Antidependency (WAR): $2>4,2<10,2>6,3><, 4 \times 6,4>6,0<6,8<10$

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c3) out-of-order order execution, with renaming;


|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 4 |
| Cycle 2 | 2 | 5 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.



## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c3) out-of-order order execution, with renaming;


True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.

Antidependency (WAR): $2>4,2<10,2>6,3><, 4 \times 6,4>6,0<6,8<10$

## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c3) out-of-order order execution, with renaming;

$$
\begin{aligned}
& \text { 1: } \quad \mathrm{R} 1 \leftarrow 100 \\
& \text { 2: } \quad \mathrm{R} 5 \leftarrow \mathrm{R} 1+\mathrm{R} 2 \\
& \text { 3: } \quad R 7 \leftarrow R 5+1 \\
& \text { 4: } \quad \mathrm{R} 11 \leftarrow \mathrm{R} 2+\mathrm{R} 4 \\
& \text { 5: } \quad \mathrm{R} 51 \leftarrow 0 \\
& \text { 6: } \quad \mathrm{R} 21 \leftarrow \mathrm{R} 4-25 \\
& \text { 7: } \quad \text { R3 } \leftarrow \text { R7-2 } \\
& \text { 9: } \mathrm{R} 10 \leftarrow 0
\end{aligned}
$$

|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 4 |
| Cycle 2 | 2 | 5 |
| Cycle 3 | 3 | 6 |
| Cycle 4 | 7 | 9 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7-8.


## Problem 3

c. Consider a superscalar computer on which the execution of each instruction takes one cycle; the computer has two arithmetic units. Show how instructions are executed in consecutive cycles with: c3) out-of-order order execution, with renaming;

$$
\begin{array}{ll}
\text { 1: } & \text { R1 } \leftarrow 100 \\
\text { 2: } & \text { R5 } 5 \text { R1 + R2 } \\
\text { 3: } & \text { R7 } \leftarrow \text { R5 + 1 } \\
\text { 4: } & \text { R11 } \leftarrow \text { R2 + R4 } \\
\text { 5: } & \text { R51 } \leftarrow 0 \\
\text { 6: } & \text { R21 } \leftarrow \text { R4 - 25 } \\
\text { 7: } & \text { R3 } \leftarrow \text { R7-2 } \\
8: & \text { R41 } \leftarrow \text { R11 + R3 } \\
\text { 9: } & \text { R10 } \leftarrow 0
\end{array}
$$

|  | ALU | ALU |
| :---: | :---: | :---: |
| Cycle 1 | 1 | 4 |
| Cycle 2 | 2 | 5 |
| Cycle 3 | 3 | 6 |
| Cycle 4 | 7 | 9 |
| Cycle 5 | 8 | 10 |

True data dependency (RAW): 1-2, 2-3, 3-7, 4-8, 4-10, 7 - 8 .


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).
b. Suppose that we are able to effectively use 17 processors rather than 9 on the parallelized portion of the code. Calculate the speedup (relative to execution on a single processor) that is achieved.

## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).

## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
a. Calculate the speedup under the aformentioned conditions (relative to execution on a single processor).


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
b. Suppose that we are able to effectively use 17 processors rather than 9 on the parallelized portion of the code. Calculate the speedup (relative to execution on a single processor) that is achieved.


## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
b. Suppose that we are able to effectively use 17 processors rather than 9 on the parallelized portion of the code. Calculate the speedup (relative to execution on a single processor) that is achieved.

$$
0.25 * T * 9
$$



## Problem 4

An application program is executed on a nine-processor cluster. The program took time $T$ on this cluster. Further, it was found that $25 \%$ of $T$ was time in which the application was running simultaneously on all nine processors. The remaining time, the application had to run on a single processor.
b. Suppose that we are able to effectively use 17 processors rather than 9 on the parallelized portion of the code. Calculate the speedup (relative to execution on a single processor) that is achieved.


$$
\begin{aligned}
& S^{17}=\frac{T_{S}}{T_{P}^{17}}=\frac{3 T}{T_{P}^{17}} \\
& T_{P}^{17}=0,25 \times \frac{9}{17} \times T+0,75 \times T=0,88 T \\
& \\
& S^{17}=\frac{3 T}{0,88 T}=3,4
\end{aligned}
$$

