Instruction pipelining

- In instruction pipeling, multiple instructions are executed in a single processor clock cycle. But in non-pipeline architectures, instructions are executed in sequence, one after another.
- Suppose there are "N" instructions with K-stages and the maximum time delay from one instruction stage to another is T time units. The difference in time required to execute an instruction with and without a pipeline is

Instruction pipeline

(K+N-1)T is the time required to execute n instructions in a pipeline.

K-number of stages

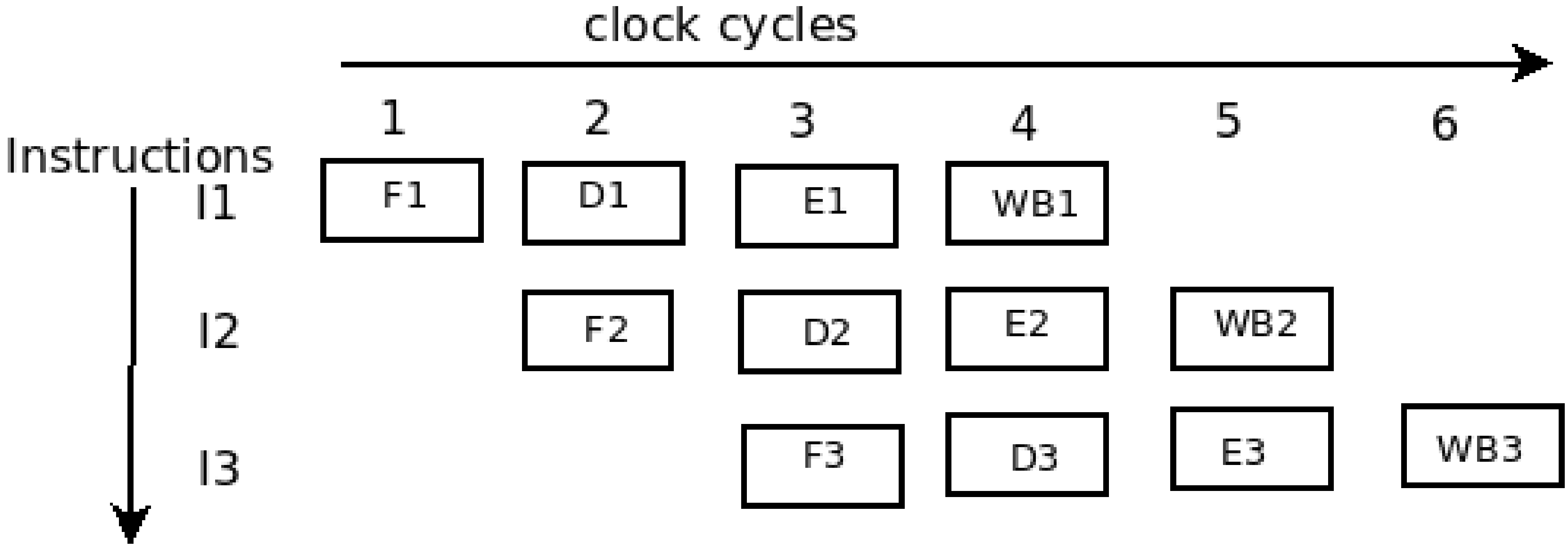
N-total number of instructions

T-time delay from one state to another

NKT is the time required to execute instructions in a nonpipeline.

Speed up factor =NK/(K+N-1)

Instruction pipeline



clock cycles

Instructions	1	2	3	4	5	6
I1	F1	D1	E1	WB1		
I2		F2	D2	E2	WB2	
I3			F3	D3	E3	WB3

Figure 5 Instruction execution in pipelining

Pipeline hazards

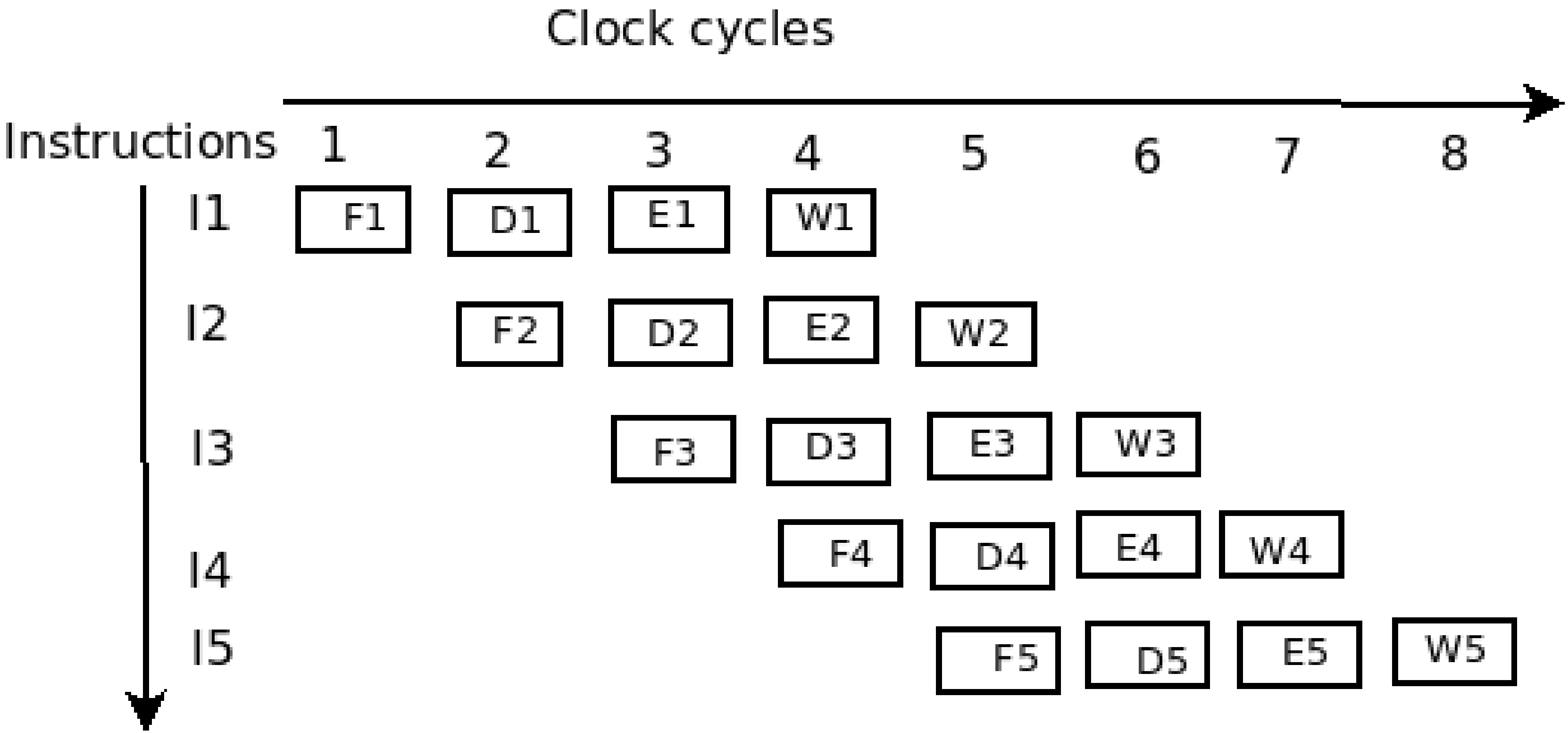
There are 3 types of pipeline hazards that occur in pipeline instruction executions.

- Resource or structural hazards
- Data hazards
- Control hazards

Resource or structural hazards

- In this hazard, if multiple instructions are in the pipeline, Assume that main memory has a single port through which this processor has to perform read or write operations one at a time. Suppose two or more instructions need the same resource (main memory) in a single processor clock cycle.

Resource or structural hazards



Clock cycles

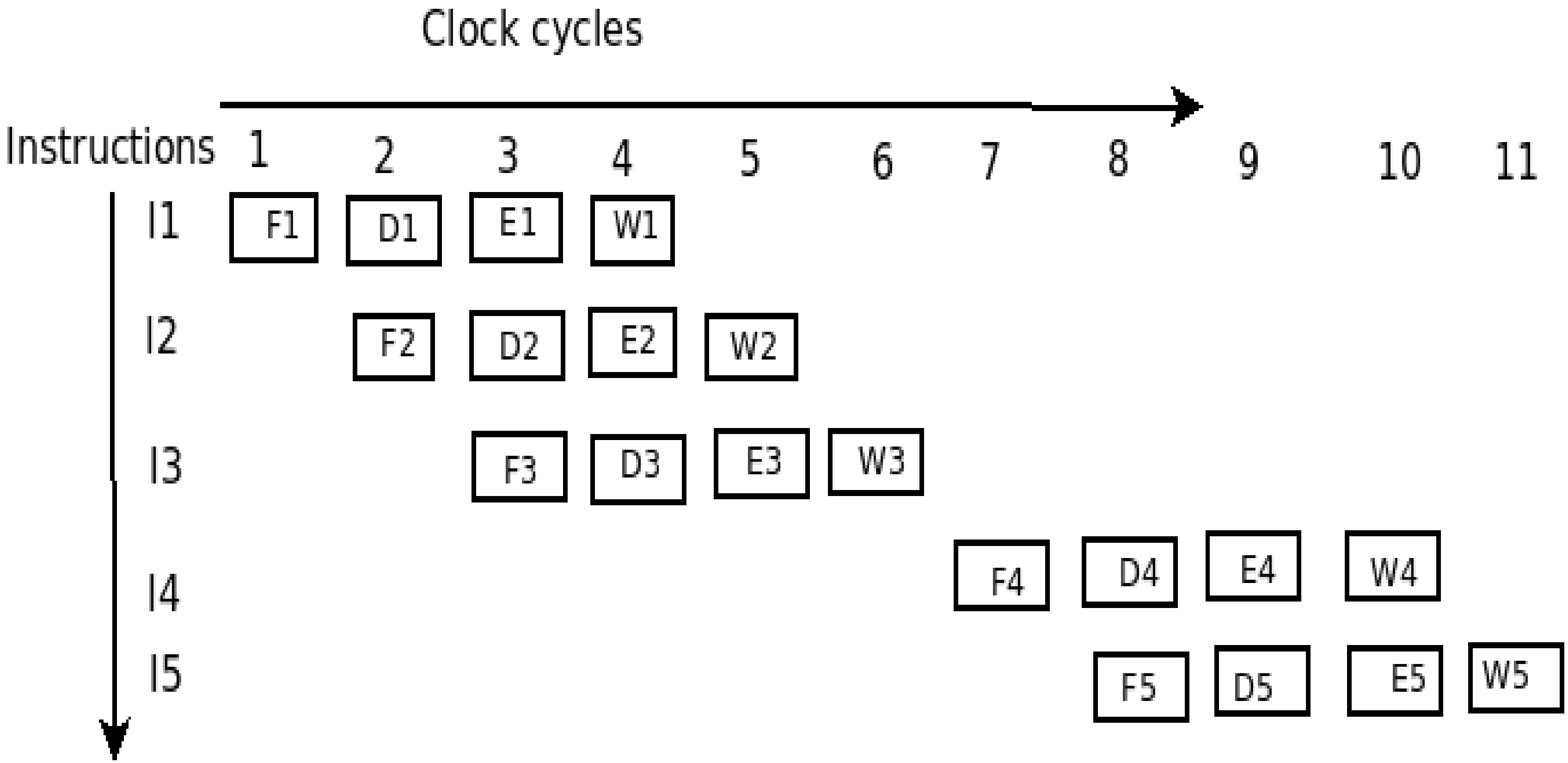
Instructions	1	2	3	4	5	6	7	8
I1	F1	D1	E1	W1				
I2		F2	D2	E2	W2			
I3			F3	D3	E3	W3		
I4				F4	D4	E4	W4	
I5					F5	D5	E5	W5

Resource or structural hazards

- Instruction I1 writes operations and I4 fetches operations in the processor's 4th clock cycle. Instruction I2 write operations and I5 fetch operations in the processor's 5th clock cycle. In the 4th clock cycle and 5th clock cycle, main memory clashes occur. This is a problem in the pipeline if more than two instructions perform different operations on the same resource (main memory).

Resource hazards solution

- To avoid the above problem, after completion of the first 3 instructions, the next instructions must be executed in a sequence.

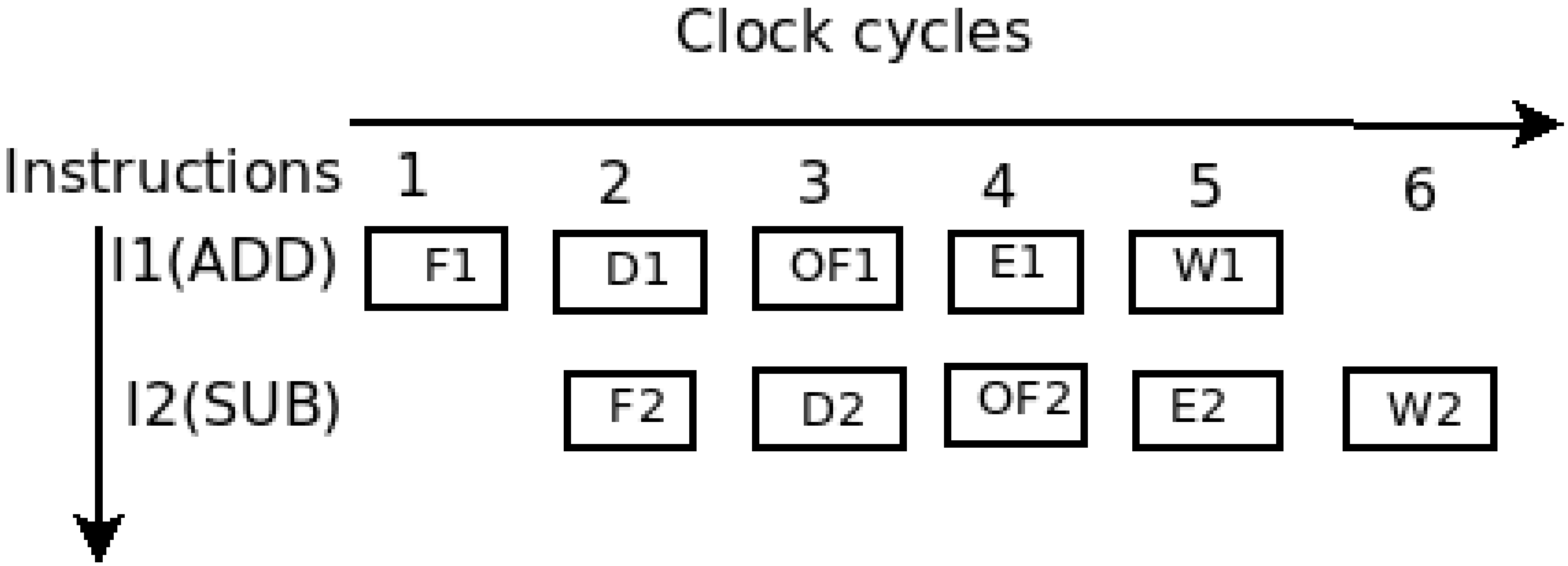


Data hazards

- Data hazards occur if two instructions are in the pipeline and are executed parallly. In the event that data hazards occur, the second instruction execution is dependent on the first instruction result.
- If instructions are executed in a sequence, then there is no problem and the program will give correct results. If both instructions are executed in the pipeline and the programme is dependent on the previous result, the programme will produce incorrect results.
- Example:

```
ADD EAX, EBX
SUB ECX, EAX
```

Data hazards



Instructions	1	2	3	4	5	6
I1(ADD)	F1	D1	OF1	E1	W1	
I2(SUB)		F2	D2	OF2	E2	W2

Figure 8 Instruction pipeline Data Hazard

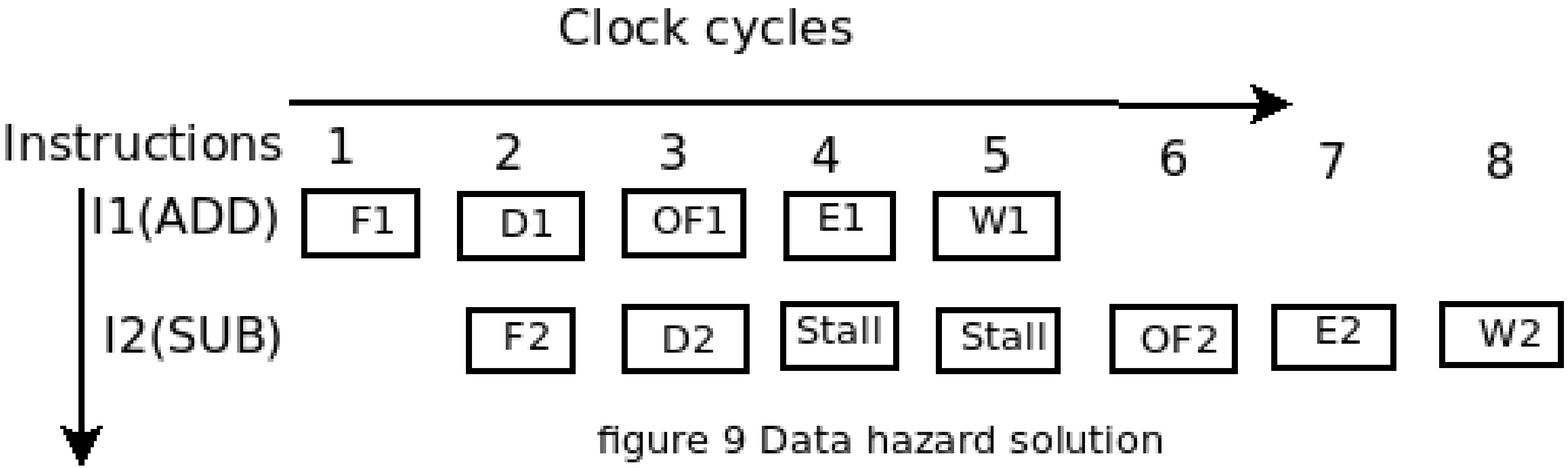
Data hazards

- Instruction I1 successfully executes after 5 clock cycles, but second instruction I2 (sub) fetches no modified operands in the 4th clock cycle. In this situation, the program will give an incorrect result.

Data hazard solution

- During execution of instruction I1 for I2, the operand is not fetched from the previous instruction. After completion of instruction I1, instruction I2 fetched the operand in the 6th clock cycle. So, in clock cycles 4 and 5, instruction execution is idle. This is called pipeline stall.

Data hazard solution



Clock cycles

Instructions	1	2	3	4	5	6	7	8
I1(ADD)	F1	D1	OF1	E1	W1			
I2(SUB)		F2	D2	Stall	Stall	OF2	E2	W2

figure 9 Data hazard solution

Control hazards

- Control hazards occur for branch instructions. The processor will not know which instructions are branch instructions until after the decoding of instructions.
- Suppose there are 3 instructions(I1, I2 & I3) in the pipeline. Assume Instruction I2 is a branch instruction, but the processor knows after the decode stage of the instruction.
- In PC branch address loaded, the processor jumps to the branch instruction and starts execution.

Control hazards

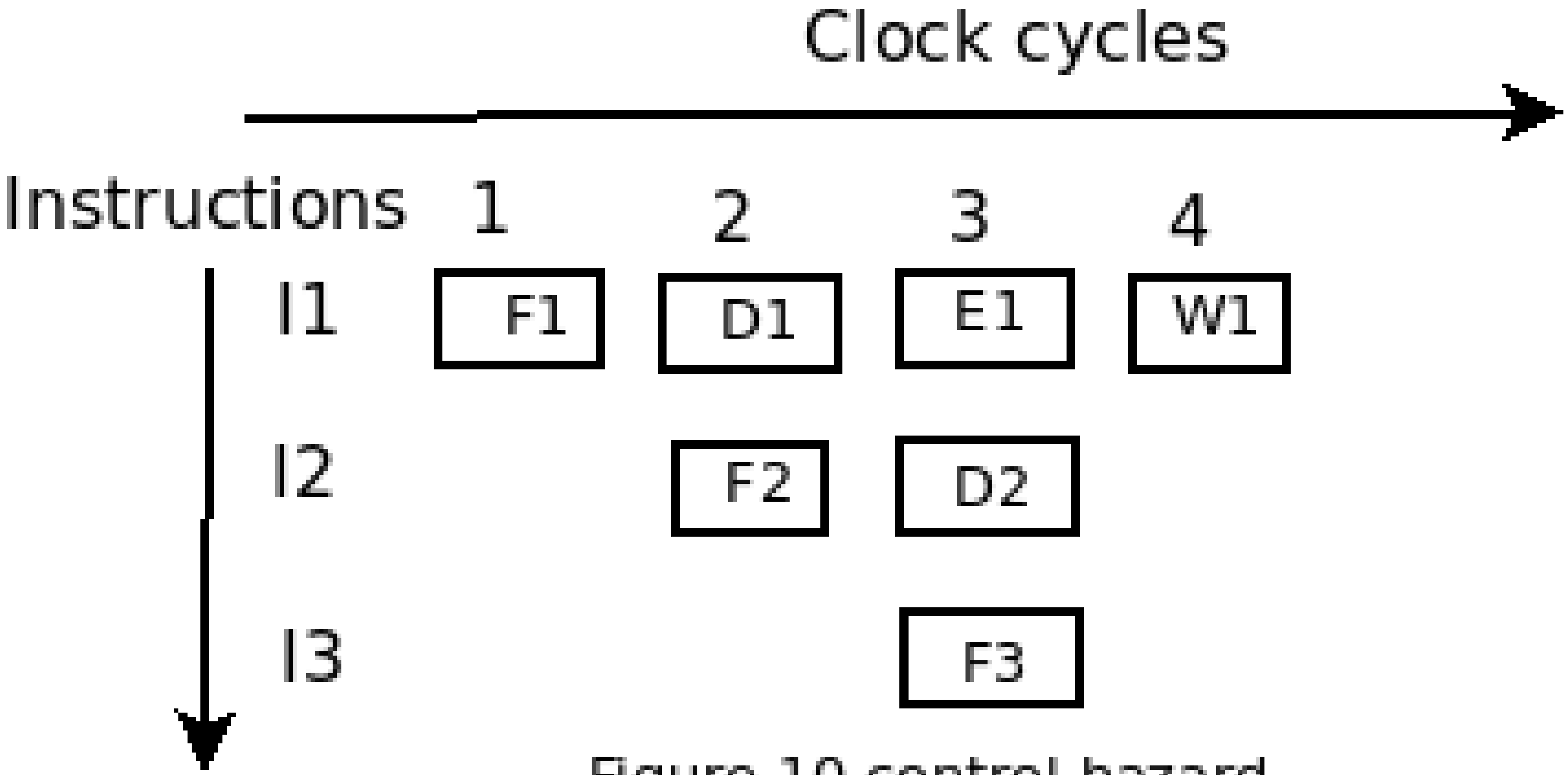


Figure 10 control hazard

Control hazard solution

- Instuctions I1, I2 and I3 in the pipeline. Instruction I2 is decoded in the 3rd clock cycle and the processor knows it is a branch instruction, but during the decoding of the I2 instruction, the I3 instruction is fetched from main memory.
- **Solution:** The instruction I3 fetch operation was completely terminated from the pipeline prior to loading the branch instruction address into PC.

Control hazard solution

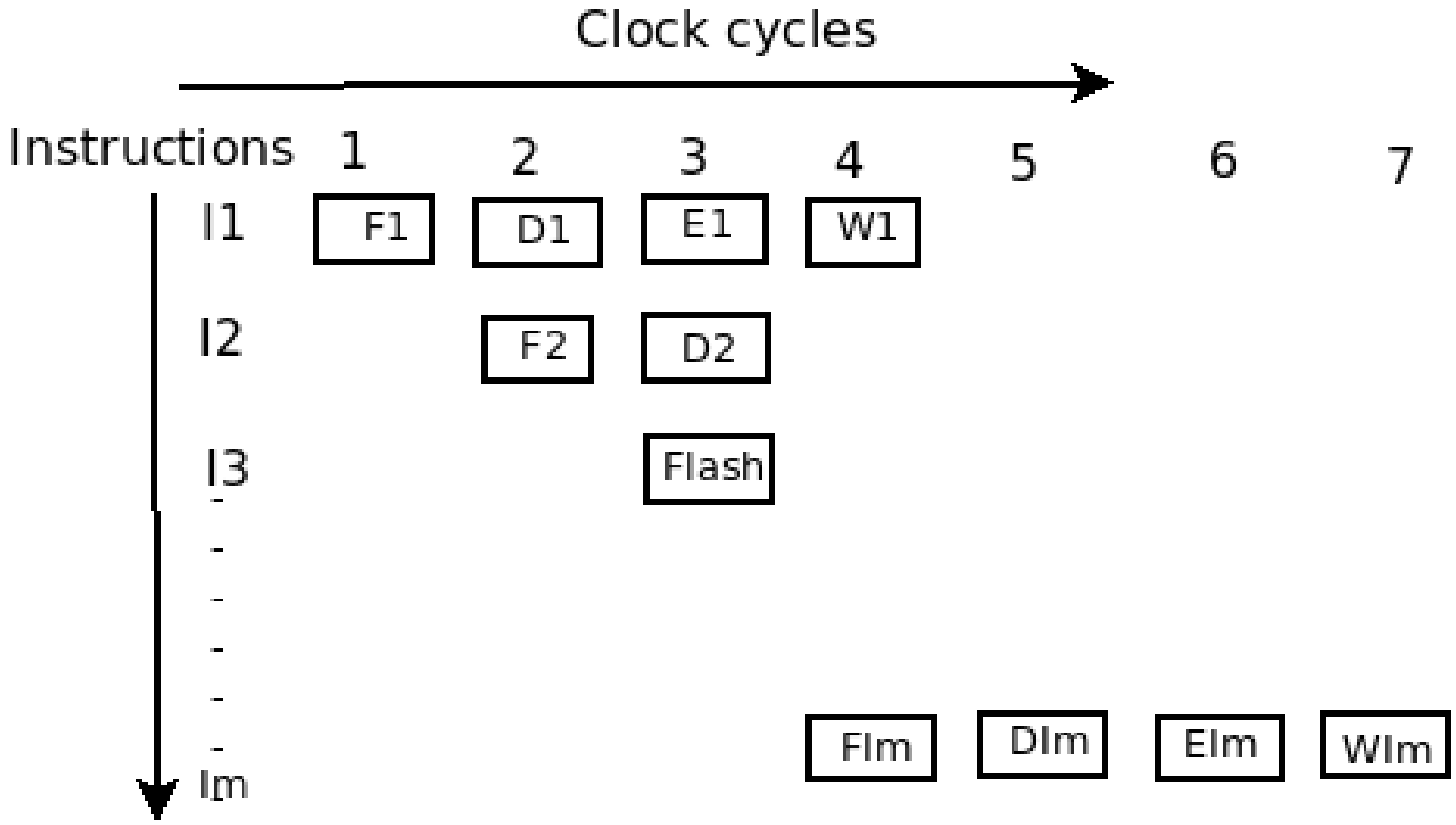


figure 11 control hazard solution

Data hazards solution