Chapter2补充题

1. Consider a processor with 64 registers and an instruction set of size 15. Each instruction has five distinct fields, namely, opcode, two source register identifiers, one destination register identifier, and a 12-bit immediate value. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 100 instructions, what is the amount of memory (in bytes) consumed by the program text?

Solution:

Number of registers=64

Number of bits to address register=$\lceil \log_2 64 \rceil$=6 bits

Number of instructions=15

Opcode size=$\lceil \log_2 15 \rceil$ =4

Opcode(4)	Reg1(6)	Reg2(6)	Reg3(6)	Immediate(12)

Chapter2补充题

1. (ctd.)

Solution: (ctd.)

Total bits per instruction=4+6×3+12 = 34 bits

Total bytes per instruction = 4.25 bytes

Due to byte alignment we can not store 4.25 bytes, without wasting 0.75 bytes.

So, total bytes per instruction = 5

Total number of instructions = 100

Total size = number of instructions × Size of an instruction

= 100 × 5 = 500 bytes

Chapter 2 补充题

2. Design a 16-bit instruction format using expanding opcode for the following: 14 three-address instructions, 30 two-address instructions, 30 one-address instructions, 32 zero-address instructions. Assume that there are 16 registers and operands are placed in register.

Solution:

16 registers→4-bit address field for one operand in the instruction

(1) 3-address instruction format

4-bit opcode 0000-1101, 12-bit address field for 3 register address.

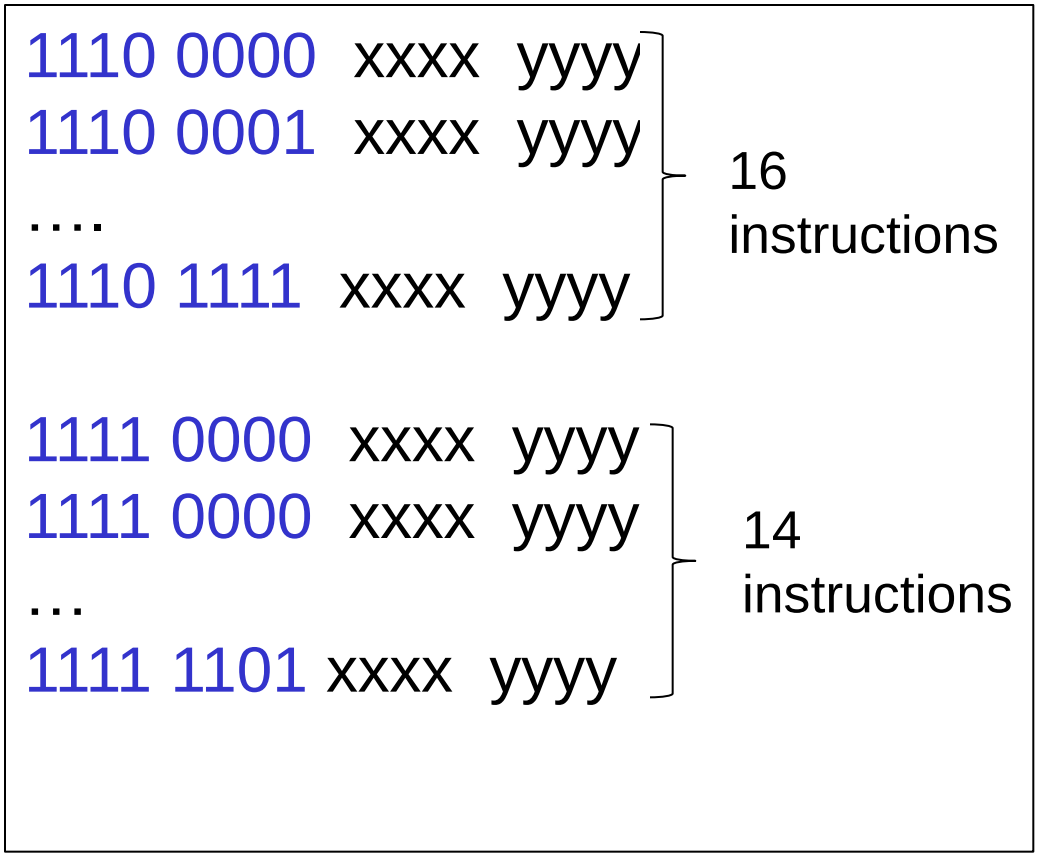
```
0000  xxxx yyyy zzzz
0001  xxxx yyyy zzzz
…
1101  xxxx yyyy zzzz
```

14 instructions

Chapter 2 补充题

(2) 2-address instruction format:

8-bit opcode 11100000-11101111(16条),11110000-11111101(14条), 8-bit address field for 2 register address.

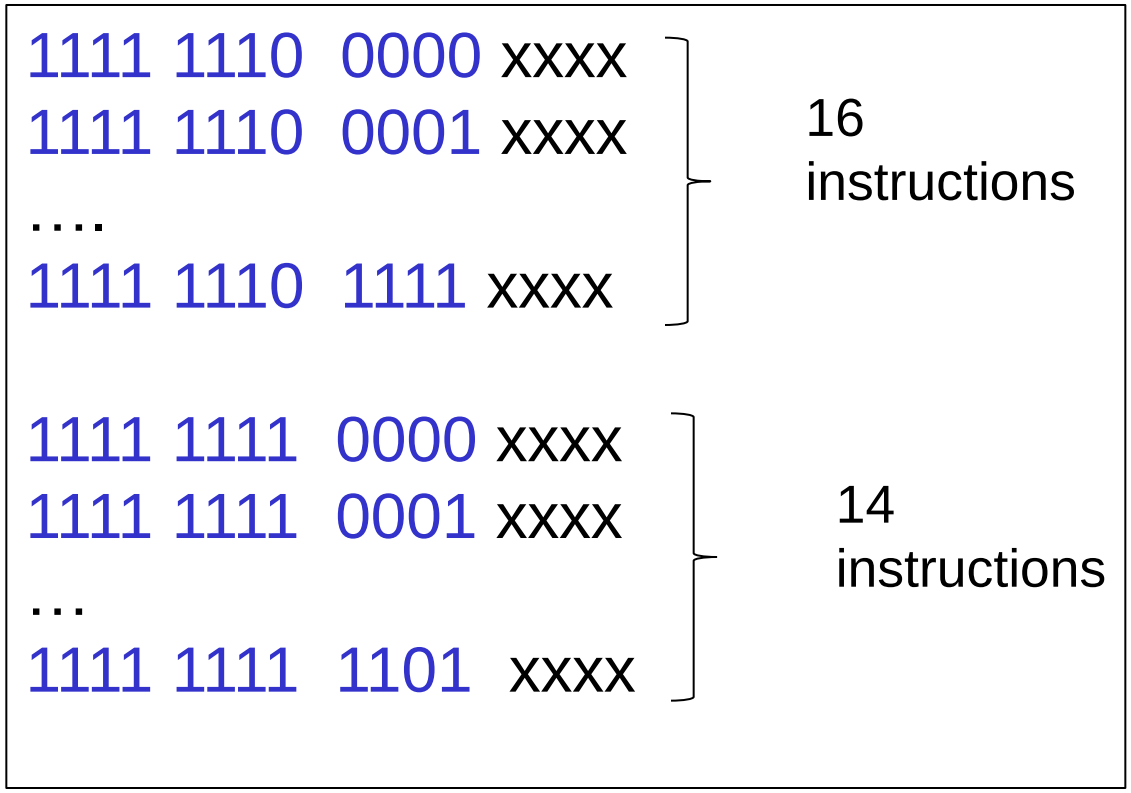


30 instructions

Chapter 2 补充题

(3) 1-address instruction format:

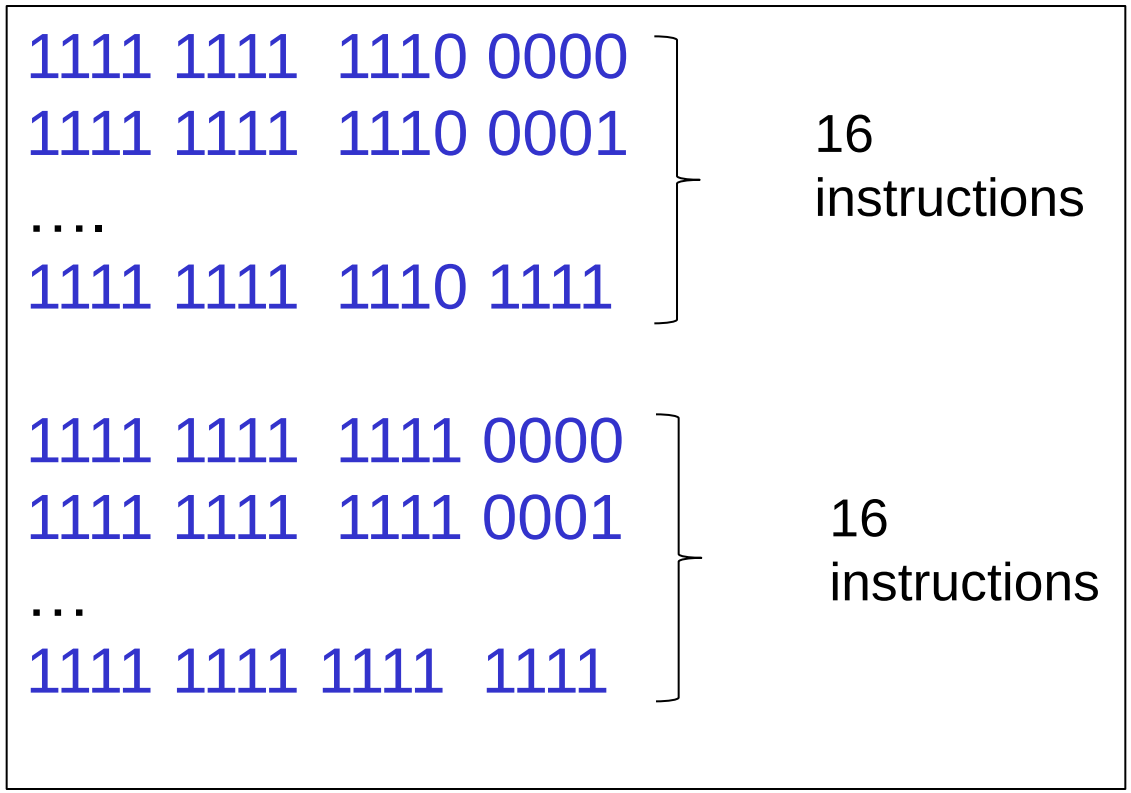
12-bit opcode 111111100000-111111101111(16条), 111111110000-111111111101(14条), 4-bit address field for 1 register address.



Chapter 2 补充题

(4) 0-address instruction format:

16-bit opcode 1111111111100000-1111111111101111(16条), 1111111111110000-1111111111111111(16条)



Chapter 5 5.4 Solution

Register contents are read in step 2 and loaded into the inter-stage registers at the end of that clock period.

下表列出的是每个周期开始时的各寄存器值

Step	RA	RB	RZ	RY	R6
3	1000	2500	*	*	7500
4	1000	2500	−1500	*	7500
5	1000	2500	−1500	−1500	7500
1	1000	2500	−1500	−1500	−1500

* These values are determined by the previous instruction.

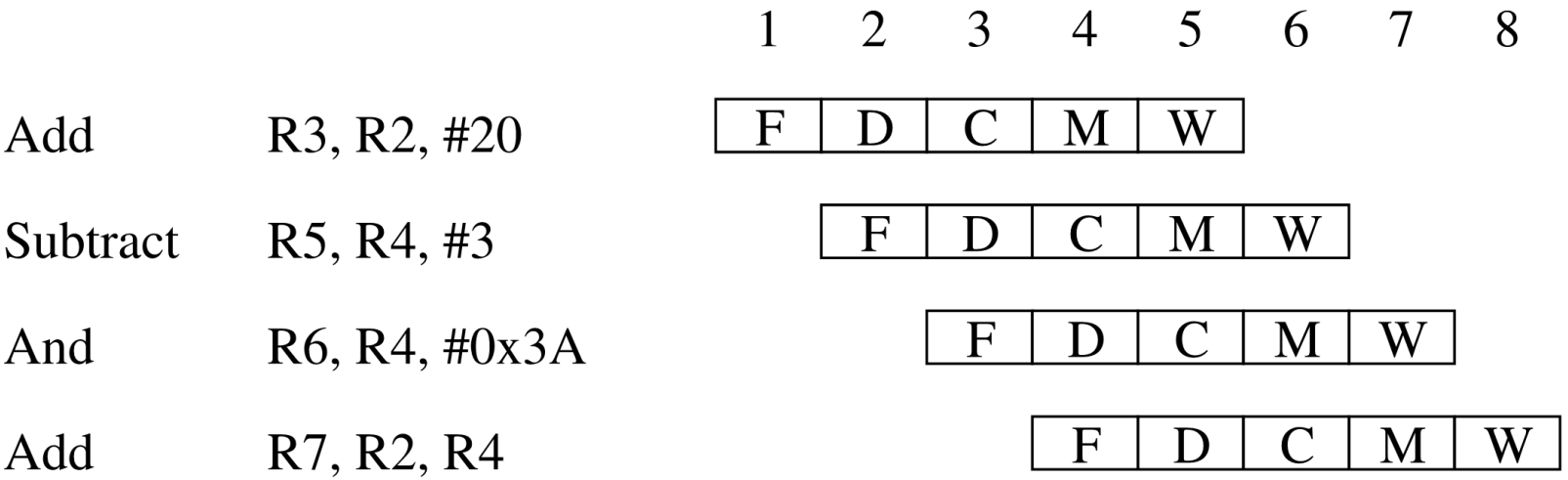
5.4 Solution (ctd.)

下表列出的是每个周期结束时的各寄存器值

Step	RA	RB	RZ	RY	R6
3	1000	2500	-1500	*	7500
4	1000	2500	-1500	-1500	7500
5	1000	2500	-1500	-1500	-1500
1	1000	2500	-1500	-1500	-1500

Chapter 6 6.1 Solution

(a) The pipeline execution diagram for the given code is shown below.



The description of activity in each stage during each cycle is given below. Activity for instructions prior to the first Add instruction and instructions after the second Add instruction is not described, but would nonetheless occur as determined by the type of instruction and the stage of the pipeline.

6.1 Solution

(a)

Cycle	Stage	Activity
1	F	fetching Add instruction
2	F	fetching Subtract instruction
	D	decoding Add instruction, reading register R2 (value 2000)
3	F	fetching And instruction
	D	decoding Subtract instruction, reading register R4 (value 50)
	C	performing arithmetic $2000 + 20 = 2020$ for Add instruction
4	F	fetching Add instruction
	D	decoding And instruction, reading register R4 (value 50)
	C	performing arithmetic $50 - 3 = 47$ for Subtract instruction
	M	no operation for Add instruction
5	D	decoding Add instruction, reading register R2 (value 200) and register R4 (value 50)
	C	performing logic operation $50 \wedge 3\mathrm{A}_{16} = 50$ for And instruction
	M	no operation for Subtract instruction
	W	write result of 2020 for Add instruction to register R3
6	C	performing arithmetic for Add instruction
	M	no operation for And instruction
	W	write result of 47 for Subtract instruction to register R5
7	M	no operation for Add instruction
	W	write result of 50 for And instruction to register R6
8	W	write result of 2050 for Add instruction to register R7

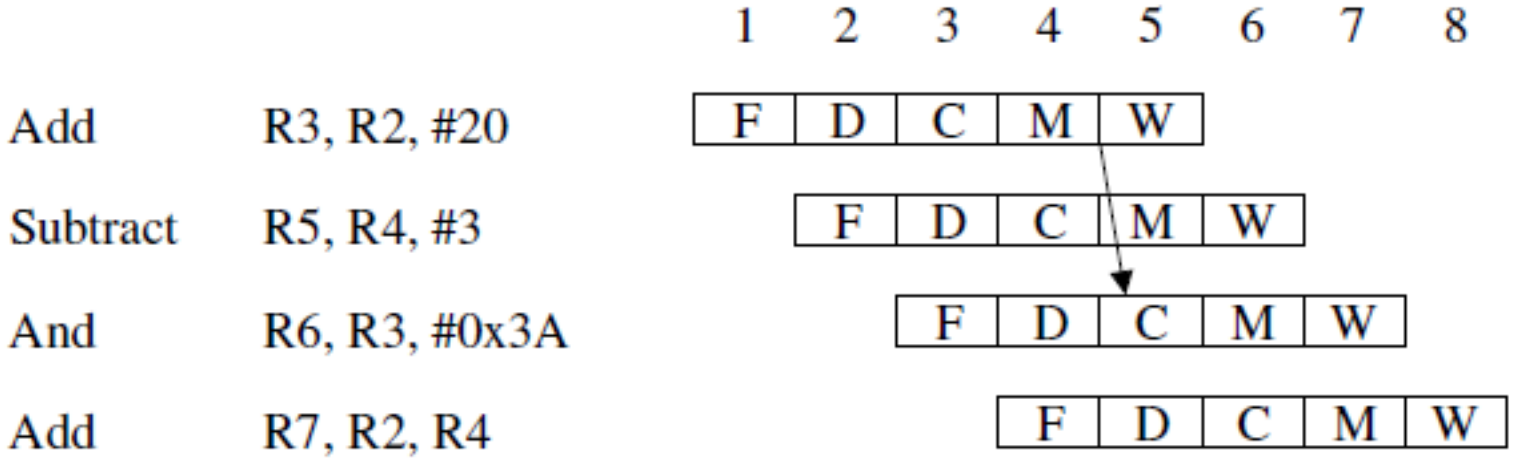
6.1 Solution

(*b*) The contents of each register during each cycle are described in the table below. The notation '–' is used to indicate that there is insufficient information to determine the register contents in that cycle. The instruction occupying the IR in each cycle is identified symbolically with an abbreviation.

	2	3	4	5	6	7	8
IR	Add	Sub	And	Add	–	–	–
PC	1004	1008	1012	1016	1020	1024	1028
RA	–	2000	50	50	2000	–	–
RB	–	–	–	–	50	–	–
RZ	–	–	2020	47	50	2050	–
RY	–	–	–	2020	47	50	2050

6.2 Solution

(a) The pipeline execution diagram for the given code is shown below.



Cycle	Stage	Activity
1	F	fetching Add instruction
2	F	fetching Subtract instruction
	D	decoding Add instruction, reading register R2 (value 2000)
3	F	fetching And instruction
	D	decoding Subtract instruction, reading register R4 (value 50)
	C	performing arithmetic $2000 + 20 = 2020$ for Add instruction

6.2 Solution

(a)

Cycle	Stage	Activity
4	F	fetching Add instruction
	D	decoding And instruction, *reading register R3 (value unknown)*
	C	performing arithmetic $50 - 3 = 47$ for Subtract instruction
	M	no operation for Add instruction
5	D	decoding Add instruction, reading register R2 (value 200) and register R4 (value 50)
	C	performing logic operation $2020 \wedge 3A = 32$ for And instruction
	M	no operation for Subtract instruction
	W	write result of 2020 for Add instruction to register R3
6	C	performing arithmetic for Add instruction
	M	no operation for And instruction
	W	write result of 47 for Subtract instruction to register R5
7	M	no operation for Add instruction
	W	write result of 32 for And instruction to register R6

6.2 Solution

(*b*) The contents of each register during each cycle are described in the table below.

	2	3	4	5	6	7	8
IR	Add	Sub	And	Add	–	–	–
PC	1004	1008	1012	1016	1020	1024	1028
RA	–	2000	*prev R3*	50	2000	–	–
RB	–	–	–	–	50	–	–
RZ	–	–	2020	47	32	2050	–
RY	–	–	–	2020	47	32	2050

The contents of RZ in cycle 4 and RY in cycle 5 are determined as follows:

$2020 \wedge 3A_{16} = 7E4_{16} \wedge 3A_{16} = 20_{16} = 32$

流水线部分补充题

Consider the following sequence of instructions being processed on the pipelined 5-stage RISC processor:

```
Load      R4, #100(R2)
Add       R5, R2, R3
Subtract  R6, R4, R5
And       R7, R2, R5
```

(1)Identify all the data dependencies in the above instruction sequence. For each dependency, indicate the two instructions and the register that causes the dependency.

Solution: There are three data dependencies in this instruction sequence

Subtract instruction depends on Load instruction for register R4

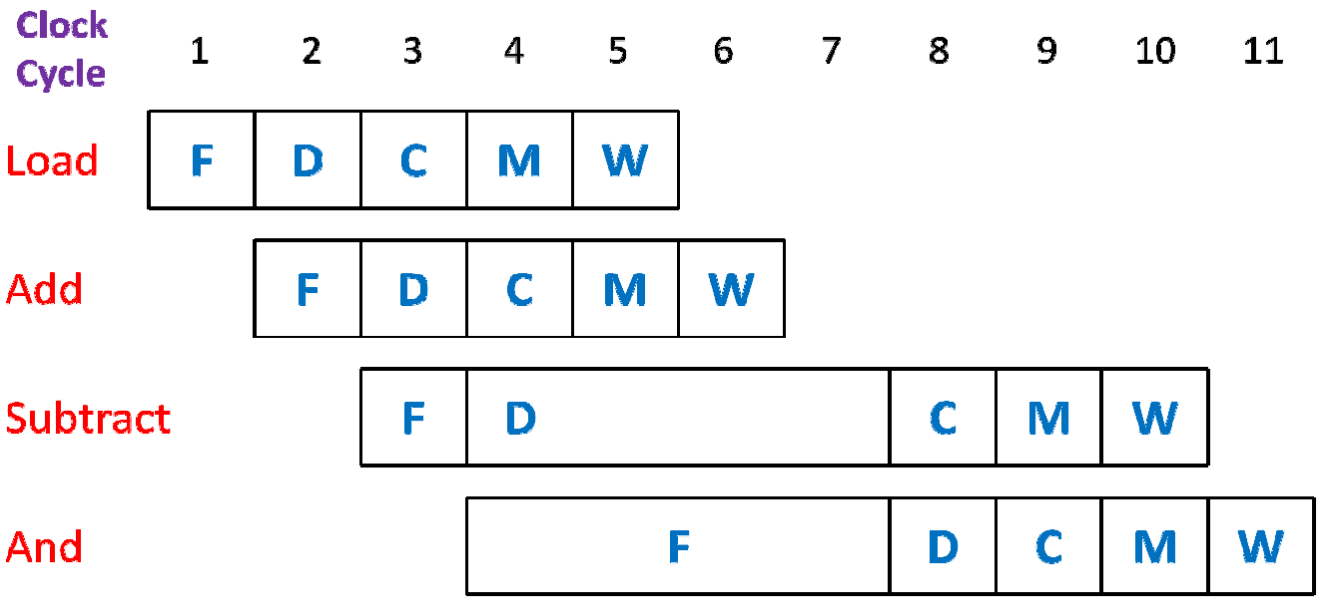
Subtract instruction depends on Add instruction for register R5

And instruction depends on Add instruction for register R5

补充题

(2)Assume that the pipeline does not use operand forwarding. Also assume that the only sources of pipeline stalls are the data hazards. Draw a diagram that represents instruction flow through the pipeline during each clock cycle.

Solution: The following diagram shows the instruction flow through the pipeline.



补充题

(3)Assume that the pipeline uses operand forwarding. There are separate forwarding paths from the outputs of stage-3 and stage-4 to the input of stage-3. Draw a diagram that represents the flow of instructions through the pipeline during each clock cycle. Indicate operand forwarding by arrows.

Solution: The following diagram shows the instruction flow through the pipeline in the presence of operand forwarding: