1.1) 21 $\xrightarrow{2}$ 000010101

-12 $\xrightarrow{2}$ 000001100 → 111110011

111110100

2's comp $\xrightarrow{}$ 111101010 + 1 = 111101011

MD: 000010101

MR: 111110100

MD

000010101

PD	MR	MX	
000000000	111110100	0	(1)

CR

1001

PD	MR	MX	
000000000	011111010	0	(2)

CR

1000

PD	MR	MX	
000000000	001111101	0	(3)

CR

0111

PD	MR	MX
111101011	001111101	0

CR

0111

(4)

PD
11111010 1

MR
10011111 0

MX
1

CR
0110

$$\begin{array}{r} 11111010 \\ +\ 00001010 \\ \hline 000001010 \end{array}$$

PD
000001010

MR
10011111 0

MX
1

CR
0110

(5)

PD
000000101

MR
0100111111

MX
0

CR
0101

$$\begin{array}{r} 000000101 \\ +\ 111101011 \\ \hline 111110000 \end{array}$$

PD
11111 0000

MR
0100111111

MX
0

CR
0101

(6)

PD
111111000

MR
0010011111

MX
1

CR
0100

(7)

PD
11111100

MR
0001001111

MX
1

CR

0011

PD	MR	MX	(8)
11111110	000010011	1	

CR
0010

PD	MR	MX	(9)
111111111	000001001	1	

CR
0001

PD	MR	MX
11111111	100001100	1

CR
0000

Result: 1111111100000100

2's comp → 0000000000111110 + 1

11111100

4 + 8 + 16 + 32 + 64 + 128 = 252.

So we verified: -12 x 21 = -252 ✓.

1.2) -5 → 000101 → 2's comp → 111010 + 1

111011

$-6 \rightarrow 000110 \rightarrow 111001 + 1 = 111010$

MD: 111011

MR: 111010-

MD
111011

PD
000000

MR
111010

CR
110

MX
0

(1)

PD
000000

MR
011101

CR
101

MX
0

(2)

PD
000101

MR
011101

CR
101

MX
0

PD
000010

MR
101110

CR
100

MX
1

(3)

$111011 + 000010 = 111101$

PD
111101

MR
101110

MX
1

CR
100

PD
11110

MR
110111

MX
0

(4)

111110
000101
000011

PD
000011

MR
110111

MX
0

CR
011

PD
000001

MR
111011

MX
1

(5)

CR
010

PD
000000

MR
111101

MX
1

(6)

CR
001

PD
000000

MR
011110

MX
1

(7).

CR
000

result: 0000000011110

$\Rightarrow \quad 2+4+8+16 = 30$.

$(-6) \times (-5) = 30$

2) opcode : 6 bits.
operand : 10 bits

1. # of different instructions $= 2^6 = 64$
(opcode determines the instruction)
2.
unsigned:
highest number $= 2^{10} - 1 = 1023$
signed:
highest number: $2^9 - 1 = 511$.

3) Register uses in pipelines:
1. Need to preserve the state of the operands as stage advance.
2. Used to keep the values of the control signals after each stage.

3. Store the results generated after each stage.

We need registers to store the state of each stage, w/o the registers the design isn't possible.

4) Description of the types of memory addressing → Check slides.

5) 1. $-35 \rightarrow 00100011 \xrightarrow{2\text{'s comp}} 11011101$

$20 \rightarrow 00010100$

$$\begin{array}{r} 11011101 \\ +\ 00010100 \\ \hline 11111001 \end{array}$$

2. $-101 \rightarrow 00011001 01 \xrightarrow{2\text{'s comp}} 11100110 11$

$31 \rightarrow 00000011111$

$$\begin{array}{r} 11100110 11 \\ +\ 00000011111 \\ \hline 11101111010 \end{array}$$

3. $512 \rightarrow 1000000000$

$256 \rightarrow 0100000000$

$$\begin{array}{r} 1100000000 \\ \hline 1 \end{array}$$

overflow

4. $-1173 \rightarrow 0100010101 10 \rightarrow 1011101010 01$

$$\begin{array}{r} 1 \\ \hline 1011101010 10 \end{array}$$

-320 → 000101000000 → 111010111111
+ 1
111011000000

+ 101110101010
111011000000
1101001101010

overflow.

6) 2 reasons CPU manufactures have different instruction sets:

1. Intellectual property: each manufacture has its own technology, so their own isfruction set might provide a market advantage
2. Make it difficult for competitors to develop when they have the majority of the market

Problem-

1. Compatibility
2. Cost of maintaining multiple vendors.

7)

CR	CPI	Inst. per sec.	
3.5×10^9 Hz	1.2	$\frac{3.5\times10^9}{1.2} = 2.9167\times10^9$	P1
2×10^9 Hz	2	$\frac{2\times10^9}{2} = 10^9$	P2
4×10^9 Hz	3.4	$\frac{4\times10^9}{} = 1.176\times10^9$	P3

3.4

1. P_1 has the highest instructions per second

2.

	# inst. in 12 sec.	clocks in 12 sec.
P_1	35×10^9	4.2×10^{10}
P_2	12×10^9	2.4×10^{10}
P_3	14.12×10^9	4.8×10^{10}

3. time $= 0.8 \times 12 = 9.6$

	CPI	inst. per sec.	CR
P1	1.38	$\frac{35 \times 10^9}{9.6} = 3.645 \times 10^9$	$3.645 \times 10^9 \times 1.38 = 5.03$ GHz
P2	2.3	$\frac{12 \times 10^9}{9.6} = 1.25 \times 10^9$	$1.25 \times 10^9 \times 2.3 = 2.875$ GHz
P3	3.91	$\frac{14.12 \times 10^9}{9.6} = 1.47 \times 10^9$	$1.47 \times 10^9 \times 3.91 = 5.751$ GHz.

8)

ADD $r1, $r2, $r3 $r1 ← $r2 + $r3.

1. At the end of cycle 5:
   read $2, $6
   written $8
2. 5th cycle → 5th instruction is being loaded. In the same cycle the result of 1st instruction is being stored. Since the only potential hazard is between instruction 1 and 5, the forwarding unit is not doing anything.

a) Allows to make a more efficient use of all the hardware by dividing the hw into individual group of components that can execute a different instructions at the same time.
Makes execution faster.
An example, check the laundry scenario presented on the book.