It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is Ox B0000000. It is NOT overflow, because MSB of x5 and x6 are the same.		No.	
(a) CPI for P2: 2×10% + 2×20% + 2×20% = 2 (b) Class A: 1.0×106×10% = 1×105 instructions Class B. C. D. have 2×105, 5×105, 2×105 instructions Clock Cycles for P1: x105x1 + 2×105x2 + 5×105x3 + 2×105x3 = 2.6×105 Clock Cycles for P2: x106x2 = 2×106 (c) Time cost of P1: 2×106 = 1.04 × 10-3 s Time cost of P2: 2×106 = 1.04 × 10-3 s Time cost of P2: 3×107 ≈ 6.67×10-4 s Hence P2 is faster. 2. (a) x30 is [0x50000000]. It is [0x800000000]. It is [0x80000000]. It is [0x8000000]. It is [0x80000000]. It is [0x8000000]. It is [0x800000	Homework 2	Da.	te
(a) CPI for P2: 2×10% + 2×20% + 2×20% = 2 (b) Class A: 1.0×106×10% = 1×105 instructions Class B. C. D. have 2×105, 5×105, 2×105 instructions Clock Cycles for P1: x105x1 + 2×105x2 + 5×105x3 + 2×105x3 = 2.6×105 Clock Cycles for P2: x106x2 = 2×106 (c) Time cost of P1: 2×106 = 1.04 × 10-3 s Time cost of P2: 2×106 = 1.04 × 10-3 s Time cost of P2: 3×107 ≈ 6.67×10-4 s Hence P2 is faster. 2. (a) x30 is [0x50000000]. It is [0x800000000]. It is [0x80000000]. It is [0x8000000]. It is [0x80000000]. It is [0x8000000]. It is [0x800000	1. (a) Global CPI for P1: 1×10%	+ 2x 20%+ 3x 50%+3x 20%=	2.6
(b) Class A: 1.0×106×10% = 1×105 instructions Class B. C. D. have 2×105, 5×105, 2×105 instructions Clock Cycles for P1: x105×1+2×105×2+5×105×3+2×105×3 = 2.6×105 Clock Cycles for P2: x106×2 = 2×105 (c) Time cost of P1: 2-5×107 = 1.04×10-35 Time cost of P2: 2×106/3×107 ≈ 6.67×10-45 Hence P2 is faster. 2. (a) x30 is [0x50000000] It is [0x50000000] It is [0x80000000] It is [0x8000000] It is [0x80000000] It is [0x8000000]	Global CPI for P2: 2×10%+2	2× 20% + 2× 50% + 2× 20% = 2	27
Class B. C. D have 2×10 ⁵ , 5×10 ⁵ , 1×10 ⁵ instructions Clock Cycles for P1: x10 ⁵ x1+2×10 ⁵ x2+5×10 ⁵ x3+2×10 ⁵ x3 = 2.6×10 ⁶ Clock Cycles for P2: x10 ⁶ x2 = 2×10 ⁶ C) Time cost of P1: 2.5×10 ⁷ = 1.04×10 ⁻³ S Time cost of P2: 2×10 ⁶ / _{3×10⁷} ≈ 6.67×10 ⁻⁴ S Hence P2 is faster. 2. (a) x30 is [0×50000000] It is [overflow], because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is [0×B0000000] It is [NOT overflow], because MSB of x5 and x6 are the same. 2. (a) x30 is [0×B0000000] It is [12] in signed 8-bit: 201100000 = 100001112, MSB of 23 and 112 are both 0. (a) 000101112+011100000 = 100001112, MSB of 23 and 112 are both 0. but MSB of vesult is 1, overflow. Saturating arithmetic yesult in [127] in decimal. (b) 000101112-011100000 = 0001011114+100100000 = 101001116, MSB of 2001011114-01110000000000000000000000000	(b) Class A: 1.0×106×10% = 1×105	instructions	4
Clock Cycles for P1: x105x1+2x105x2+5x105x3+2x105x3 = 2.6x106 Clock Cycles for P2: x106x2 = 2x106 C) Time cost of P1: 2.5x107 = 1.04x10-3 s Time cost of P2: 2x106 = 6.67x10-4 s Hence P2 is faster. 2. (a) x30 is 0x50000000. It is 0verflow , because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is 0xB0000000.] It is WOT overflow , because MSB of x5 and x6 are the same. 2. 3 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001113 , MSB of 23 and 112 are both 0. but MSB of vesult is 1, overflow. Saturating arithmetic yesult in 127 in decimal. (b) 000101112 - 011100002 = 000101113 + 100100002 = 101001113 , MSB of 23 and 112 are both 0. 000101113 - 011100002 = 000101113 + 100100002 = 101001113 , MSB of 23 and 112 are both 0. 000101113 - 011100002 = 000101113 + 100100002 = 101001113 , MSB of 23 and 112 are both 0. 000101113 - 011100002 = 000101113 + 100100002 = 101001113 , MSB of 23 and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow. Saturating arithmetic yesult and 112 are both 0, no overflow.			
(c) Time cost of P1: \(\frac{2.5 \times 10^4}{2.5 \times 10^7} = 1.04 \times 10^{-3} \text{ S}\) Time cost of P2: \(\frac{2 \times 10^4}{2 \times 10^7} \text{ \$\infty} \text{ \$\infty} 6.67 \times 10^4 \text{ S}\) Hence \(\frac{P2}{I}\) is faster. 2. (a) \(\times 30\) is \(\times 5.00000000\). It is \(\times \text{ overflow}\), because MSB of \(\times 5\) and \(\times 6\) are both 1, but MS of \(\times 30\) is \(\times 0.000000000\). It is \(\times \text{ OX B00000000}\). It is \(\times \text{ NOT overflow}\), because MSB of \(\times 5\) and \(\times 6\) are the same. 112 in signed 8-bit: \(\text{ o11100000}\). 123 in signed 8-bit: \(\text{ o11100000}\). 124 in signed 8-bit: \(\text{ o11100000}\). 125 in decimal. 127 in decimal. (b) \(\text{ o0010111}\). \(\text{ o11100000}\). = \(\text{ o0010111}\). \(\text{ host of 23}\) and \(\text{ o11100000}\). \(\text{ overflow}\). \(\text{ Saturating arithmetic result only 112 are both 0, no overflow. Saturating arithmetic result and \(\text{ o112}\) are both 0, no overflow. Saturating arithmetic result and \(\text{ o112}\) are both 0, no overflow. Saturating arithmetic result	12		6×106
Time cost of P1: 2.5×108 = 1.04×103 s Time cost of P2: 2×106 ≈ 6.67×104 s Hence P2 is faster. 2. (a) x30 is [0x50000000] It is [overflow], because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is [0x80000000] It is [NOT overflow], because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 (a) 00010112+01100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in [127] in decimal. (b) 000101112-011100002 = 000101112+ 100100003 = 101001113, MSB of 22 and 112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	Clock Cycles for PZ: 1×106×2		ř.
Hence P2 is faster. 2. (a) x30 is 0x50000000. It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is 0x80000000. It is WOT overflow, because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 112 in signed 8-bit: 011100002 (a) 000101112 + 0111000002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001113, MSB of 200101112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	(c) Time cost of P1: 2.5×109 =		è
2. (a) x30 is [0x5000000] It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is [0x8000000] It is [WOT overflow], because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 122 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in [127] in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001113, MSB of 23 and 112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	Time cost of P2: $\frac{2\times10^6}{3\times10^7}$?	= 6.67 × 10 ⁻⁴ s	V.
It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is 0x80000000. It is NOT overflow, because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 200101112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	Hence Pz is faster.		5
It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is 0x80000000. It is NOT overflow, because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 200101112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	1 20 Line 19 19 19 19 19 19 19 19 19 19 19 19 19	1) 1900 x 30 m x 2000 (1)	1100
It is overflow, because MSB of x5 and x6 are both 1, but MS of x30 is 0, it must overflow. (b) x30 is 0x80000000. It is NOT overflow, because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 200101112 are both 0, no overflow. Saturating arithmetic result and 112 are both 0, no overflow. Saturating arithmetic result	2. (a) x30 is 0x50000000.		
of x30 is 0, it must overflow. (b) x30 is [0x80000000.] It is [NOT overflow], because MSB of x5 and x6 are the same. 23 in signed 8-bit: 000000000000000000000000000000000000	It is loverflow, because MS	B of x5 and x6 are both	1, but MSB
(b) x30 is [0x8000000] It is [WOT overflow], because MSB of x5 and x6 are the same. 23 in signed 8-bit: 000000000000000000000000000000000000			20010 0
It is NOT overflow, because MSB of x5 and x6 are the same. 23 in signed 8-bit: 011100002 112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of vesult is 1, overflow. Saturating arithmetic yesu in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic yesult	(h) v20 is [0x B000 0000]	- N	.47
23 in signed 8-bit: -0000 000001112 112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of vesult is 1, overflow. Saturating arithmetic yesu in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic yesult	14: WOT overflow, because	MSB of x5 and x6 are	the same.
112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic result		au = 1-1/2,717.75	
112 in signed 8-bit: 011100002 (a) 000101112 + 011100002 = 100001112, MSB of 23 and 112 are both 0. but MSB of result is 1, overflow. Saturating arithmetic result in 127 in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic result	10 1:4: 0000 0001	oilly assistantes	2 M 2
(a) 000101112 + 011100002 = 100001112, MSB of 25 and but MSB of result is 1, overflow. Saturating arithmetic result in [127] in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001112, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic result	3. 23 in signed 8-bit:	45, 3104 10	ed * 1
but MSB of result is 1, overflow. Saturating and 112 are both 0, no overflow. Saturating arithmetic result	112 in signed 8-bit: Officer	MCR of 23 and 11	2 are both 0.
but MSB of result is 1, overflow. Saturating and 112 are both 0, no overflow. Saturating arithmetic result	(a) 000101112 + 011100002 = 10000)[[]2 , MOD VI - + 12-0 0v.	thmetic result
in [127] in decimal. (b) 000101112 - 011100002 = 000101112 + 100100002 = 101001113, MSB of 2 and 112 are both 0, no overflow. Saturating arithmetic result	but MSB of result is 1, c	overflow. Saturaling wi	<i>0.</i> 1113.0
(b) 000101112 - 011100002 = 000101112 + 100100003 = (01001112) and 112 are both 0, no overflow. Saturating arithmetic result	the state of the s		
and 112 are both o, no overflow. Saturday	00010	1112 + 100100000 = 101001115	etic result
is [-89] in decimal.	and 112 are both o, no over	rflow. Saturating arithm	Jevis
	is [-89] in decimal.		

i	ter	multiplicand	product	operation
	0	00010000	00000000 01100100	all the second
	1		0000 0000 00110010	shift right
7	2		00000000 00011801	shift right
			0001010000011001	add multiplicand to left
-	3	1.00 7.5	0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0	shift right
L	t .		0000010100000110	shift right
5			0000001010000011	shift right add multiplicand to left
,			0001011010000011	
6			0000101101000001	shift right
			0001111101000001	add multiplicand to left
7			0000111110100000	shift right
8			0000011111010000	shift night

Hence, $0x62 \times 0x14 = 0x0700 (100 \times 20 = 2000 in decimal)$

	-0	000000	01111000000a		
5.	iter	Q	Divisor	R	operation
	0	000000	010101000000	000000111110	
		000000	001010100000	10101111110	R = R - Div R < 0, R = R + Div, shift 0 into Q shift Div Shift Div right
	2	000000	000101010000	000000111110	R=R-Div R<0, R=R+Div, shift o into Q Shift Div right
	3	000000	000010101000	000000111110	R=R-Div R<0, R=R+Div, shift 0 into Q Shift Divo right
	4	000000	000001010100	000000111110	Same as 1
	5	000000	000000 101010	000000 11110	Same as 1
	6	000001	000000010101	000000010100	R=R-Div R>0. shift 1 into Q Shift Div right
	7	000010	000000000000	000000010100	Same as 1

6. (a) S=0, exp = 000110002, frac=0, bias=127,0
num = $(-1)^{5} \times (1 + frac) \times 2^{(exp-bias)} = (-1)^{0} \times (1 + 0) \times 2^{-103} = 2^{-103} \approx 9.86 \times 10^{-32}$
(b) 63.25,0 = 111111.01, = 1.1111101 × \$\vec{1}{2}^5
S=0, exp = =5-bias = 132 = 01111010, frac = 111110100000000000000000000000000000
IEEE754 representation is 0x3D7D0000 0x427D0000.