CO: Assignment #2

Due on September 26, 2016

Group members:

Devishi Kesar(2015024)
Gaurav Gehlot(2015030)
Gunkirat Kaur(2015032)
Kushagra Arora(2015049)
Nikhil Hassija(2015065)

26/9/16 Date: CO ASSIGNMENT-2 Sr. No.

1. Interrupts service routine vs. subroutine

⇒ 1. Subroutine runs on call while ISR runs whenever a certain signal occurs.

2. We get to know where subroutine runs because we call it while we can never know where ISR will be executed.

3. Subroutine is a portion of code within a larger program, which performs a specific task and is relatively independant of the remaining code. Interrupt service routines (ISR) are to handle hardware interrupts, they are not independant threads, more like signals. ISR is called if any thread is suspended

Acall vs. Lcall.

⇒ Lcall - i) 3 byte instruction (opcode, address subroutine, target subroutine)

ii) # can be located anywhere within a 64k bytes address space.

Acall - i) It is a 2 byte instruction (Opcode, address subroutine)

ii) 11 bits used to address subroutine within 2k bytes range

Use of ACALL instead of LCALL can save ROM space

Eg:

```
ORG 0000
ST: MOV A, #40H
    MOV P1, A
    LCALL DELAY
    MOV A, #55H
    MOV P1, A
    LCALL DELAY
    SJMP ST
```

Evergreen Signature

Date : _____ Sr. No. _____

```
;Delay subroutine
  ORG 300H
DELAY: MOV R0, #0FFH
LOOP = DJNZ R0, LOOP
     RET
     END

   ORG 0000              ; ACALL
ST : SETB P0.0
     ACALL DELAY
     CLR P0.0
     ACALL DELAY
     SJMP ST
```

⇒ Street light implementation : using IE0 (External Interrupt 0)
Let LED be connected to P1.5. Usually it stays ON i.e. at night thus it is active low and ~~on~~ in presence of some light, & the LED stays on for a small time and then goes off remaining unchanged unless interupted.

```
ORG 0000
LJMP MAIN

ORG 0003H
SETB P1.5
MOV R0 , #25
INTER: DJNZ R0, INTER
   CLR P1.5
    RETI
```

Date : _____ Sr. No. _____

```
ORG 30H
MAIN : MOV IE, # 10000001B
STAY : SJMP STAY
END
```

2) External interrupts (1) Come from outside devices like from I/O devices, timing devices, power supply etc. i.e any external source
(2) Asynchronous with program
(3) Caused by an external event.
(4) Initiated by signal that occurs in the hardware of CPU.

Internal interrupts : (1) Due to illegal use of data or instruction i.e. Interrupt from within the CPU.
(2) Synchronous with the program
(3) Caused by internal error conditions such as register overflow, division by zero etc.

Maskable vs. Non maskable

Maskable Interrupts : Interrupts that can be disabled or ignored by the CPU. Eg : external hardware interrupts, timer interrupts, serial communication etc.

Non-Maskable interrupts : Interrupts that cannot be disabled or ignored by the CPU. Eg : RESET

Date : Sr. No.

⇒ Hardware vs. Software interrupts

Hardware interrupts: 1) Generated by hardware like keyboard etc.
2) Hardware interrupts handle hardware events.
3) External and internal interrupts are generated from signals that occur in the hardware of CPU

Software interrupts: 1) Generated by special software instructions i.e initiated on execution of instruction
2) Software handlers serve the application.
3) It is a special call instruction that behaves like an interrupt rather than a subroutine.

⇒ Vector vs. Non-Vectored Interrupts

→ A vectored interrupt is where the CPU actually knows the address of the ISR (interrupt service routine) in advance. All it needs is that the interrupting device sends its unique vector via a data bus and through its I/O interface to CPU, which takes this vector, checks an interrupt and then carries ISR for it. So vectored interrupt allows the CPU to know what ISR to carry out in software.

→ A non-vectored interrupt is one where which interrupting device never sends an interrupt vector. It is literally a hard coded ISR that jumps the PC to a fixed address. The CPU does not know which device caused the interrupt without polling each I/O interface in a loop and checking status of each register to find one which created interrupt.

Evergreen Signature

3

Single Precision format

Minimum number.

Exponent : 00000001
Actual Exponent = 1 - 127 = -126
Fraction : 000 . . . . 00 ⇒ significand = 1.0
Number :

$$\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$$

Maximum Number.

Exponent = 11111110
Actual Exponent = 254 - 127 = 127
Fraction : 111 . . . 11 ⇒ significand = 2.0
Number

$$\pm 2.0 \times 2^{127} = \pm 3.4 \times 10^{38}$$

Double Precision Format

Minimum Number

Exponent : 0000000000000001
Actual Exponent : 1 - 1023 = -1022
fraction : 000 . . . 00 ⇒ significand = 1.0
Number :

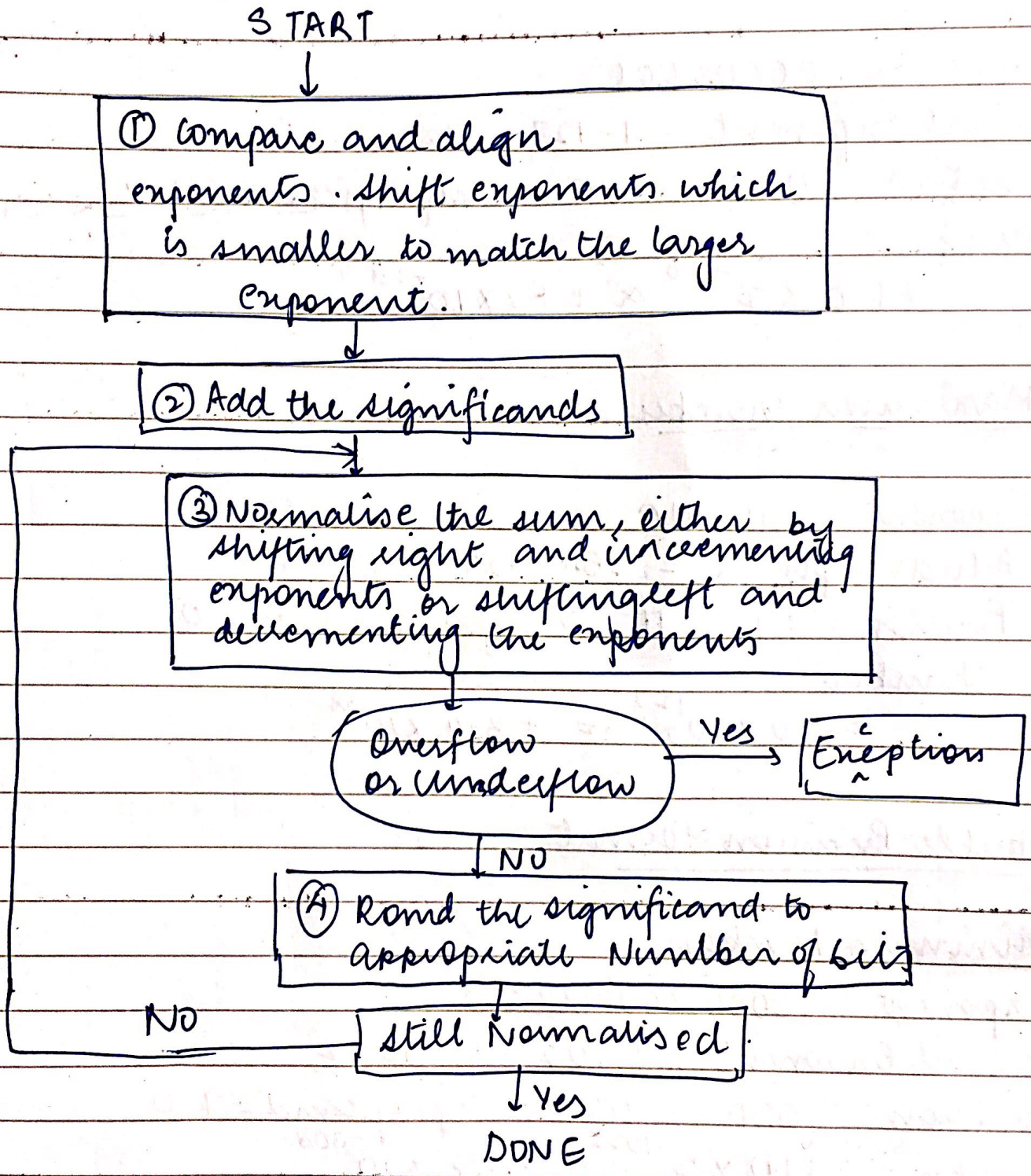
$$\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$$

Maximum Number

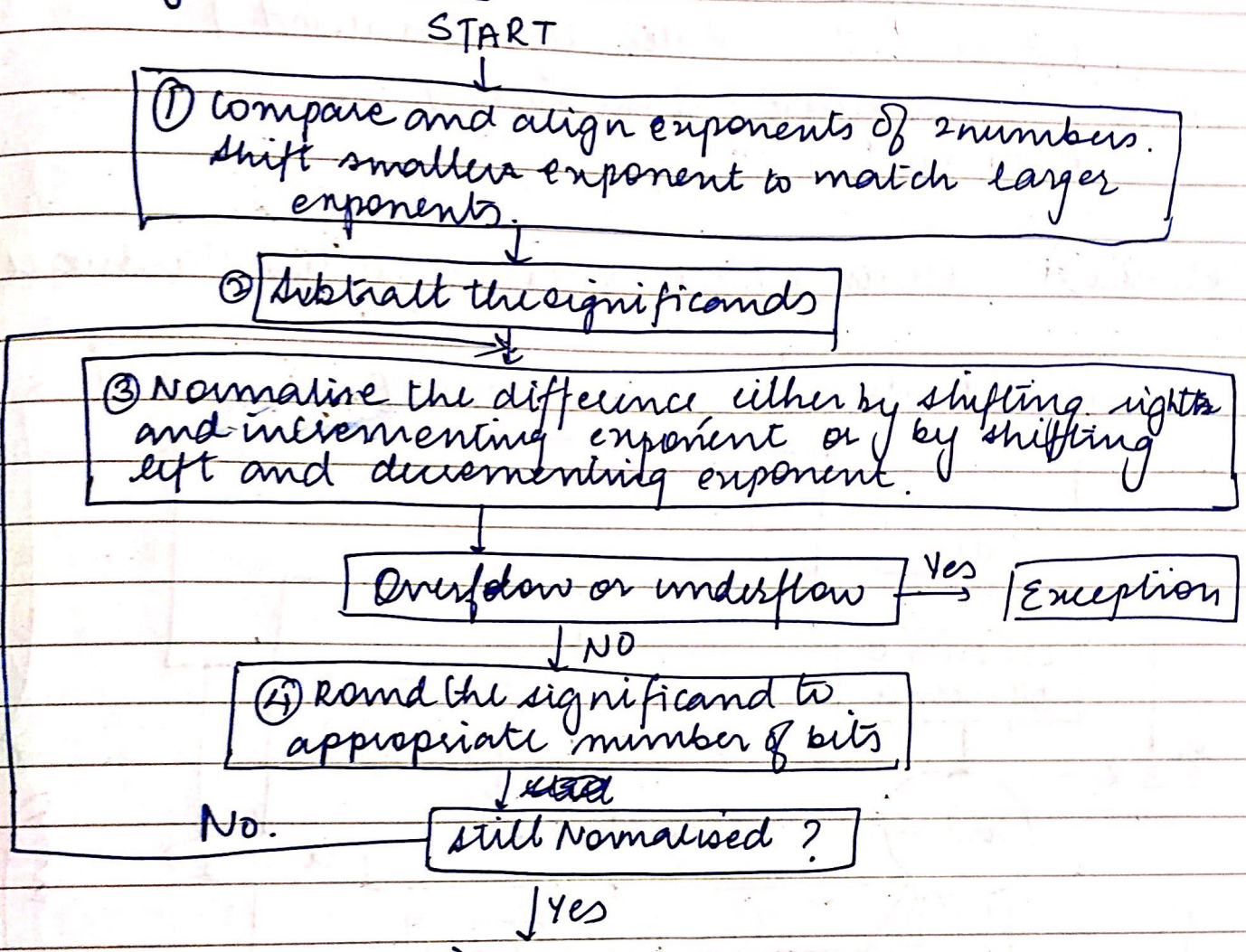
Exponent : 11111111111111110
Actual Exponent : 2046 - 1023 = 1023

Fraction: $\pm 2.0 \times 2^{1023} \approx \pm 1.8 \times 10^{308}$

Floating point addition



Floating Point subtraction



Addition

~~$9.6765 \times 10^4 + 1.769 \times 10^3$~~

$9.5213 \times 10^4 + 1.793 \times 10^3$

① Align decimal points

9.5213×10^4

$+ 0.1793 \times 10^4$ ← shift no. with smaller exponents

② Add significands.

9.5213×10^4

0.1793×10^4

9.7006×10^4

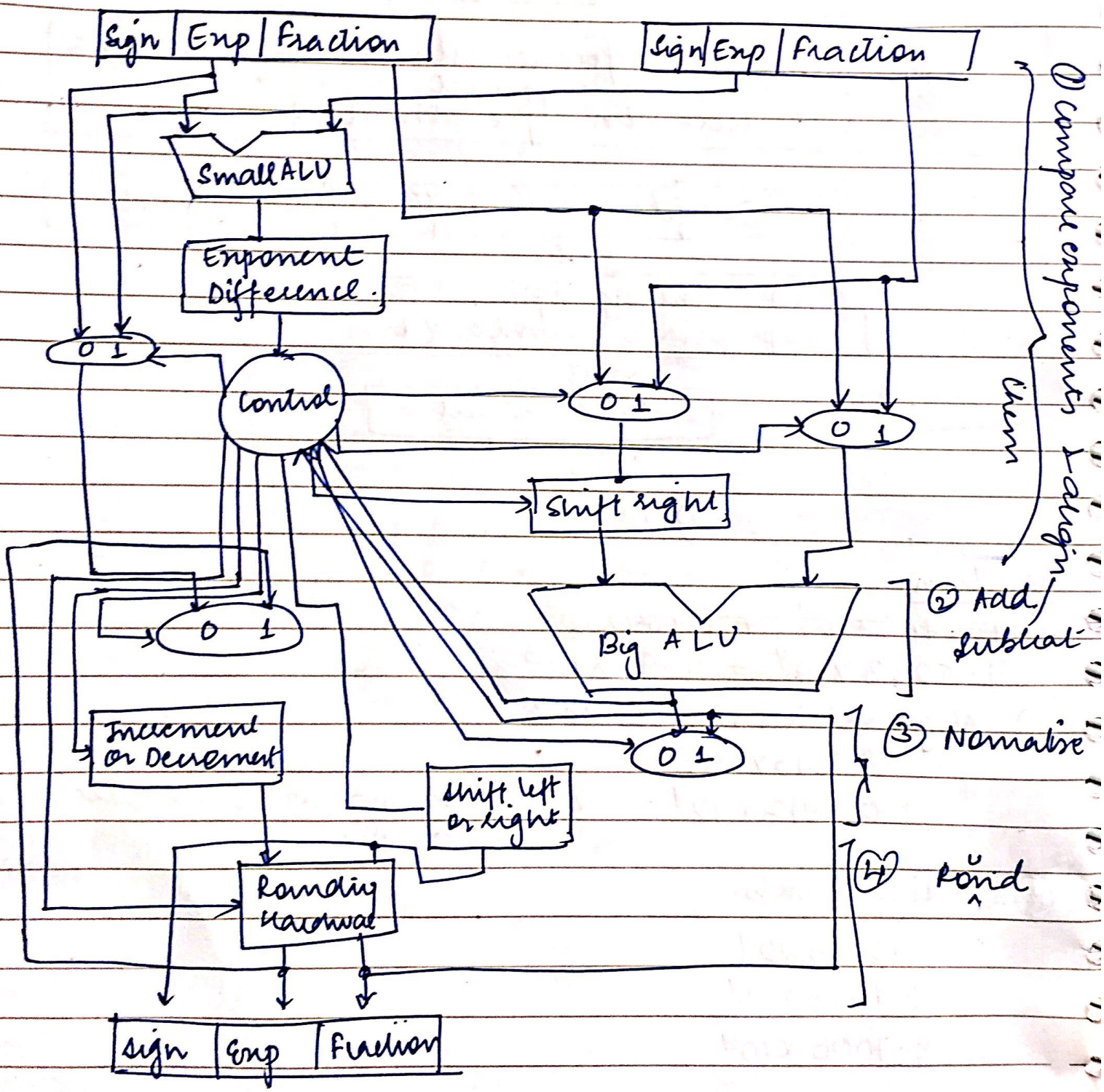
③ Normalise result

9.7005×10^4 (Already normalised)

④ Round & renormalise (if necessary)

9.701×10^4

Floating Point Adder/Subtractor Hardware



Subtraction $84.160 \times 10^{1} - 6.4 \times 10^{-1}$

① Align decimal points

$$84.160 \times 10^{1}$$
$$- 0.064 \times 10^{1}$$

② subtract significands

$$84.160 \times 10^{1}$$
$$- 0.064 \times 10^{1}$$
$$\overline{84.096 \times 10^{1}}$$

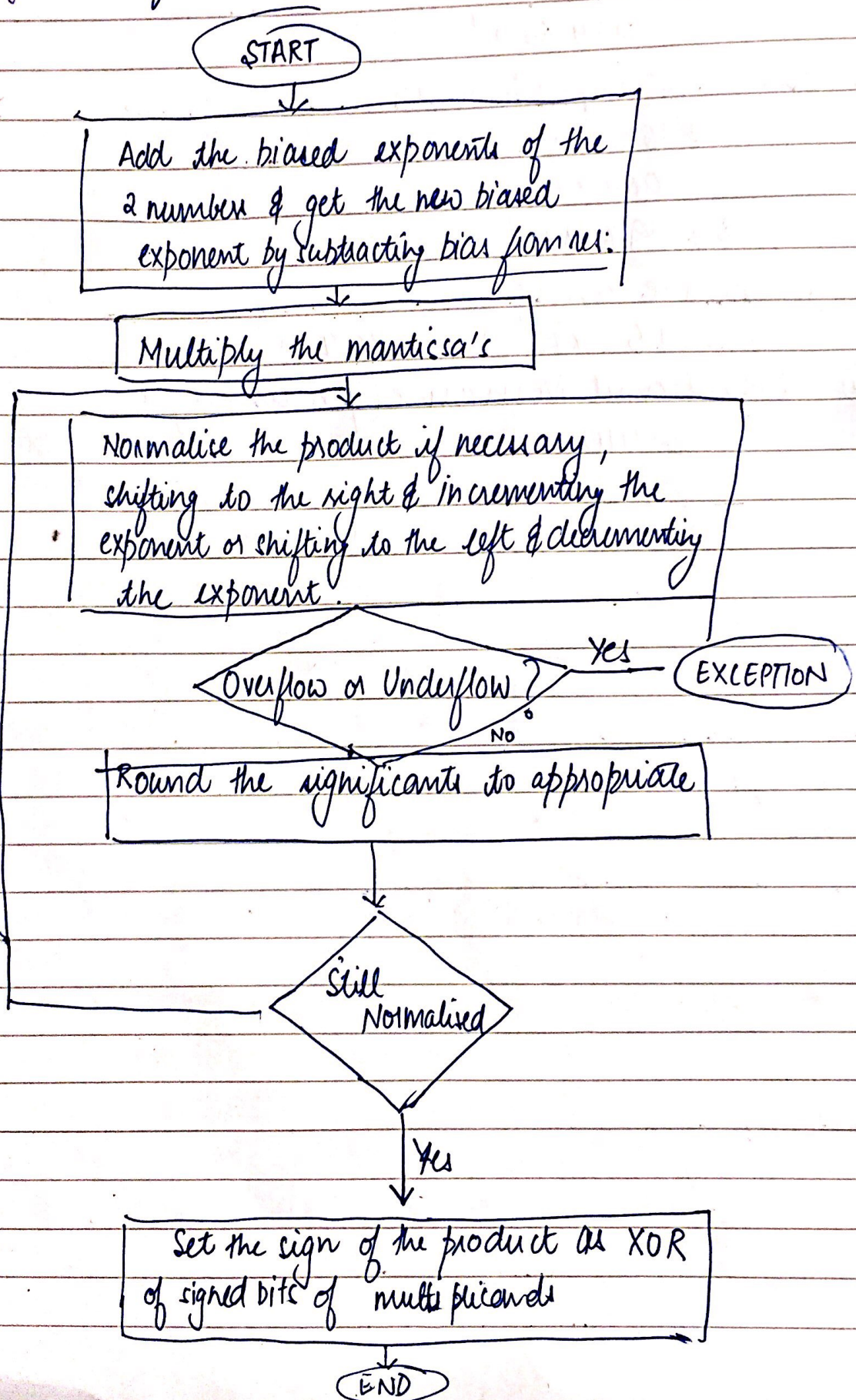
③ Normalise result

$8.4096 \times 10^{2} = 8.4096 \times 10^{2}$

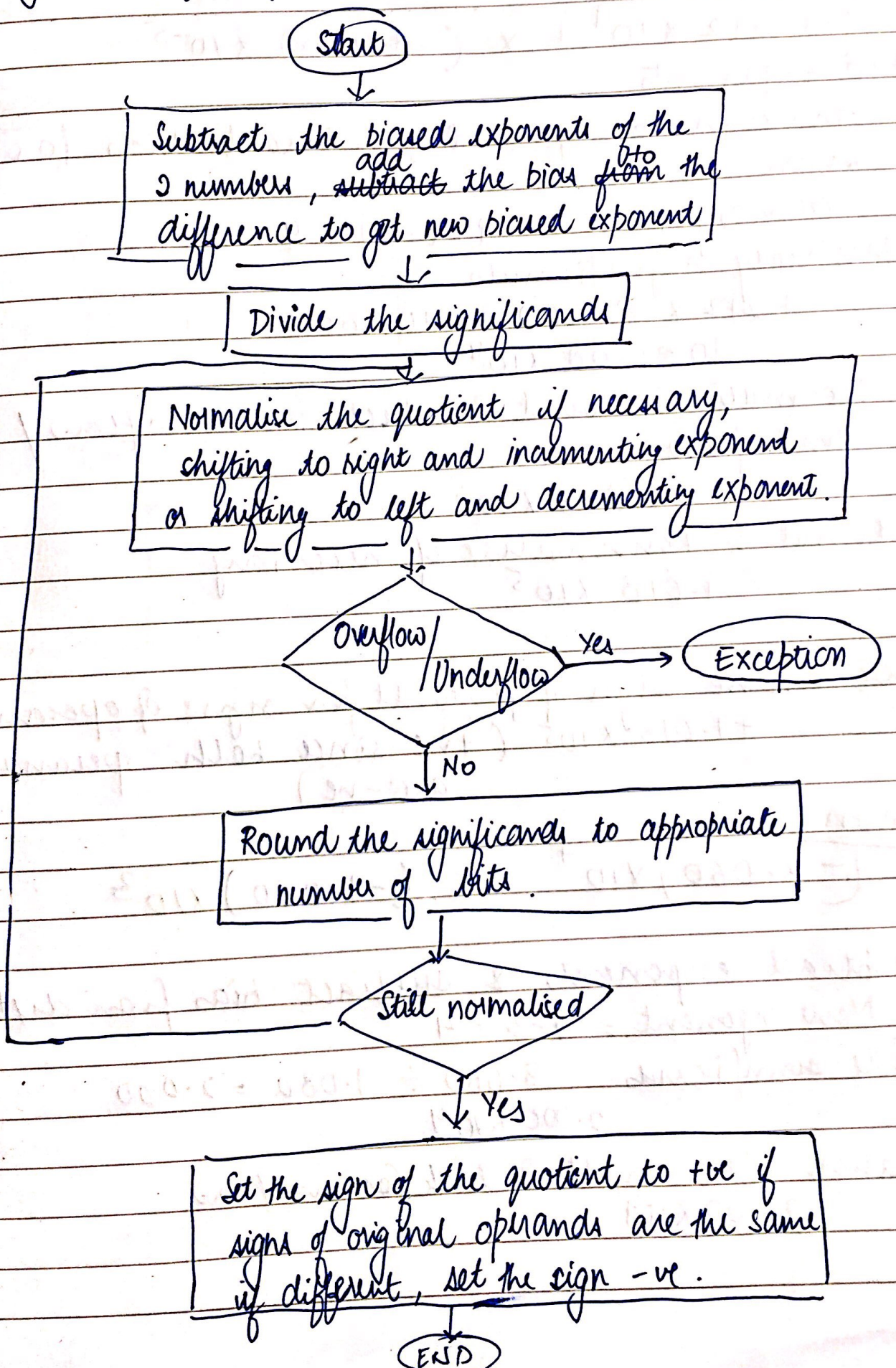
④ Round and Normalise (if necessary)

8.410×10^{2}

4. Algorithm for Floating point multiplication



Algorithm for floating point division



e.g.

Multiplication

$(-1.212 \times 10^{9}) \times (-8.400) \times 10^{-5}$

① Add exponents
For biased exponent, subtract bias for sum
New exponent = $9 + (-5) = 4$

② Multiply significands
$1.212 \times 8.400 = 10.1808$
10.1808×10^{4}

③ Normalise result & check for overflow / underflow
1.01808×10^{5}

④ Round & Renormalise if necessary
1.018×10^{5}

⑤ Determine sign of result for signs of operands
$+1.018 \times 10^{5}$ (+ve since both operands are -ve)

Division

$(-3.060) \times 10^{7} \div (-1.080) \times 10^{3}$

① Subtract exponents & subtract bias from diff
New exponent = $7 - 3 = 4$

② Divide significands $3.060 \div 1.080 = 2.000$
2.000×10^{4}

③ Normalise the result & check for overflow
2.000×10^{4}

④ Round & renormalise if necessary
2.000 x 10^4

⑤ Determine sign of result from sign of operands
2000 x 10^4 (+ve since both operands are -ve)

HARDWARE