Assignment-03

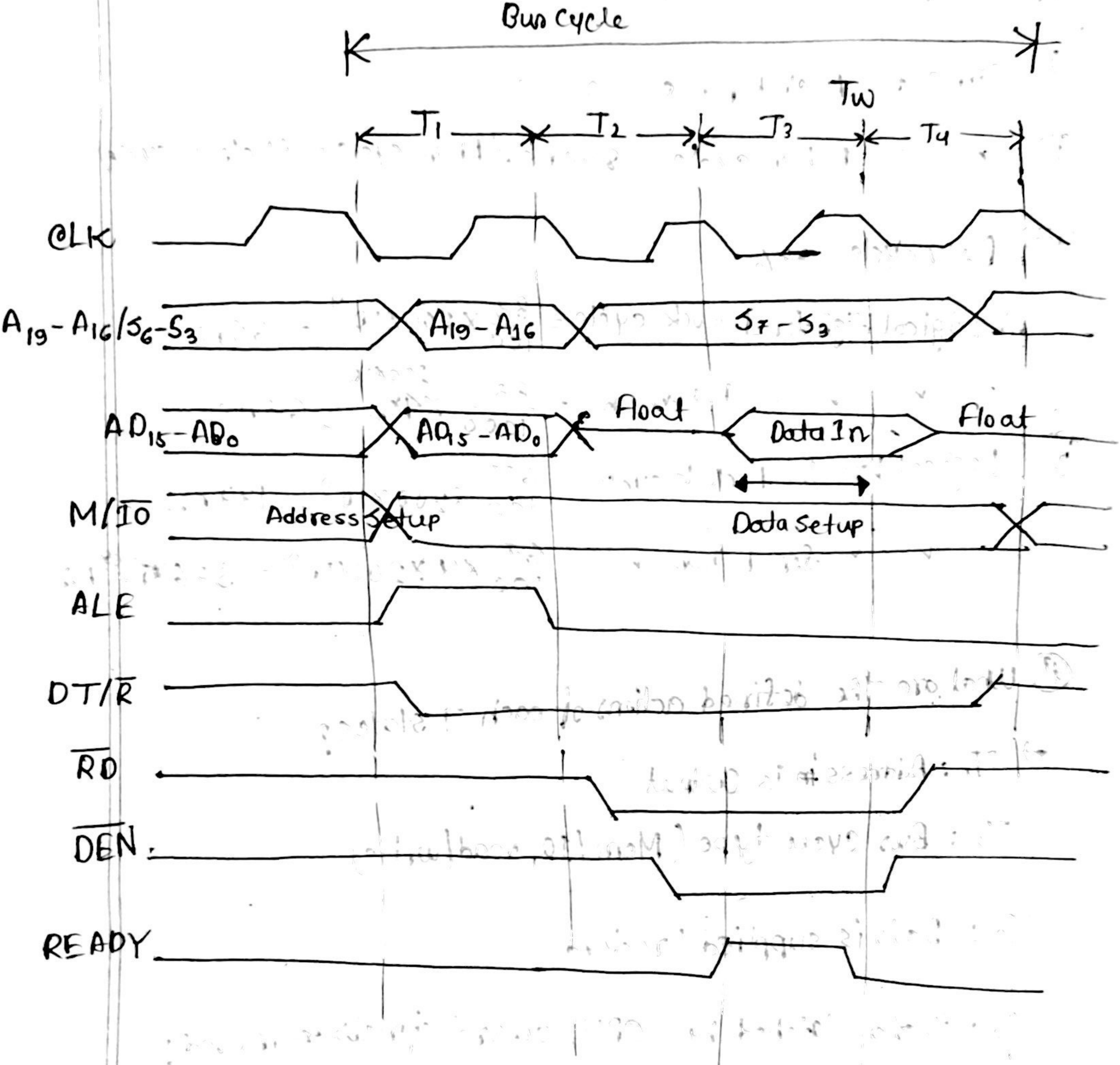
CSE341

Name: Umme Abira Azmary

ID: 20101539

Section: 10

(1) (H) Complete Bus cycle :



② If the $\overline{RD}$ and $\overline{DEN}$ is high throughout the read cycle, the data transfer will not take place. The reason is, $\overline{RD}$ and $\overline{DEN}$ are active low pin and they take part in data tranferring throughou the read cycle if they are given low value.

$\overline{RD} = \overline{0} = 1$. and $\overline{DEN} = \overline{0} = 1$.

③ If the READY pin is high while the $\overline{RD}$ pin is low, the data transfer will not take place. The reason is, when $\overline{RD}$ pin is low, it lets processor to read/write from IO/memory device, whereas the READY pin lets the microprocessor know if a memory/IO ~~pin~~ pin is activated to share. So, if the $\overline{RD}$ ~~is~~ pin is still ~~s~~ high ~~during~~ while Ready pin is activated, the data transfer will not take place.

④ If the ALE pin is low throughout the read cycle, ~~it~~ the address of whethere a data will be read~~write~~ from memory/IO will be unavailable. To be more specific, if an available address is not acknowledged by the microprocessor, ~~the~~ its corresponding data will not be read.

(5) (a) If an interrupt is caused by the INT 53 instruction, it is non-maskable. The reason is, all software generated interruption from 0-235 are non-maskable.

(b) $53 \times 4 = (212)_d$

IP $= (00D4)_h$, slot

CS $= (00D6)_h$, slot.

Now,

The starting address is $= (6B150)_h$.

(c) If a signal is received on the NMI pin, first the flag value of IF becomes 0, so that no other interrupt occurs while handling the present interrupt and the TF is also 0, so that is stops single ~~exe~~ step execution. Because, while servicing an interrupt, if single step is kept activated, it may kill much more time. And these two values of flag are pushed on the stack as MAIN PSW through stack pointer. Then, by reducing the value of stack pointer 2 the MAIN CS and by ~~reducki~~ reducing the value ~~of~~ 4, the MAIN IP value is pushed on the stack and then ISR CS and ISR IP is ~~st~~ pushed on the microprocessor.

After successful execution of interrupt, the μP pop the values of IP, CS and flags to continue the execution of the previous instruction and that's how ~~an~~ NMI interrupt is handled.

(6) (a) INTR interrupt has occured from t=0sec to 10sec.

Divide Error occurs from t=11sec to t=16sec.

So, order is INTR → Divide Error.

(b)

(b) INTR interrupt has occured from t=0 sec to t = 5sec, at t = 6 sec, Divide Error Interrupt occurs. Because of the arrival of a higher prioritized interrupt, the Divide Error Interrupt is serviced first from t= 6 sec to t = 11sec and after that, the remaining INTR is serviced from t=12 to t=16 sec.

So, order is : Divide Error → INTR

(c) INTR interrupt has occured from t = 0 sec to t = 10sec.

The single step interrupt will be serviced from t= 11sec to t = 16 sec ~~b~~ as it is a low prioritized interrupt.

So, order is : INTR → Single Step.

(d) INTR has occured from t = 0 to t = 5sec. Then, NMI, Single step and Divide Error ~~and~~ occurs at t= 6 sec. Now, the Divide Error is serviced first, then NMI, then INTR and lastly single step.

So, Order is; Divide Error → NMI → INTR → Single Step.