Name: KOTTE SAHITHI KRISHNA

Roll No: S20160010045

CPES END EXAM Question Paper No :- 2

1) PIPELINING:-

There are 5 levels of pipelining. A process takes 5 cycles in pipelining, when an instruction is fetched from the instruction memory, the decode part retrieves data from the register bank and is written into the ALU unit where the arithmetic and logical operations are performed and the memory pipeline stage reads or writes to a memory location given by a register. The writeback pipeline stage stores results in the register file.

data hazard :- The non-availability of the ~~information~~ data for an instruction which is dependent on the previous instruction in the process of pipelining is called data hazard.

a) Throughput :- The throughput increases due to pipelining. The execution time of an individual instruction does not decrease but because of the overhead in the pipeline control, it increases slightly. Throughput means the number of instructions completed in unit time. Since the instruction throughput increases the programs runs faster and has lower execution time.

Latency :- The moment an instruction is fetched and time till it is written back to register bank is the latency. Now, as there is an overhead the execution time increases and therefore increasing latency.

b) If our instructions are dependent on each other then only data hazard occurs. Data hazard is one form of pipeline hazard.

Name: KOTTE SAHITHI KRISHNA
Roll No. S20160010045

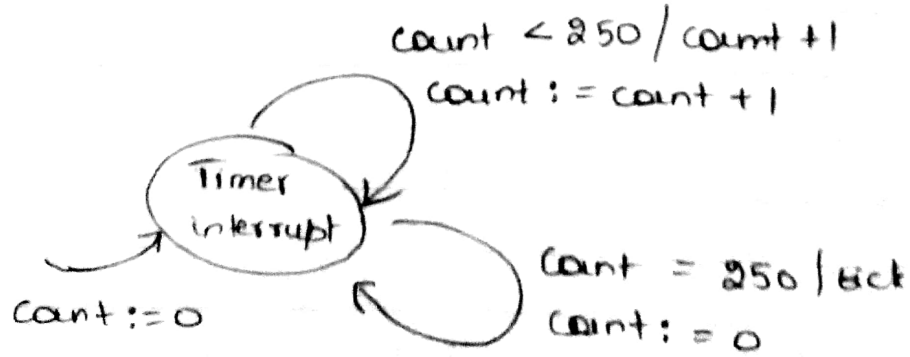
CPES END EXAM Question paper No - 2

Programmers generally expect that if instruction A is befor instruction B, then the results are available to B but in this data hazard this does not happen. There are three ways to overcome this ~~selet~~ problem.

- Explicit pipeline:- In this part the programer or compiler should deal with it, for example, if B needs data from A, the compiler must insert 3 no-operations instructions between A and B, so that it forms a pipeline bubble and enables data for B.
- Interlocks: The execution of B is delayed till the writeback stage of A has been completed, if there is a forwarding logic ie, if A directly writes the information to B then this can be reduced to 2cycles
- Out of order execution: where hardware is provided it detects a hazard, but instead of simply delaying B's execution, it starts fetching any new instruction that is independent of A or B.

If there is data hazard, it implies that the sequential instructions are independent from each other and the processor doesn't need to take care of that and writes the results in register.

2) a)



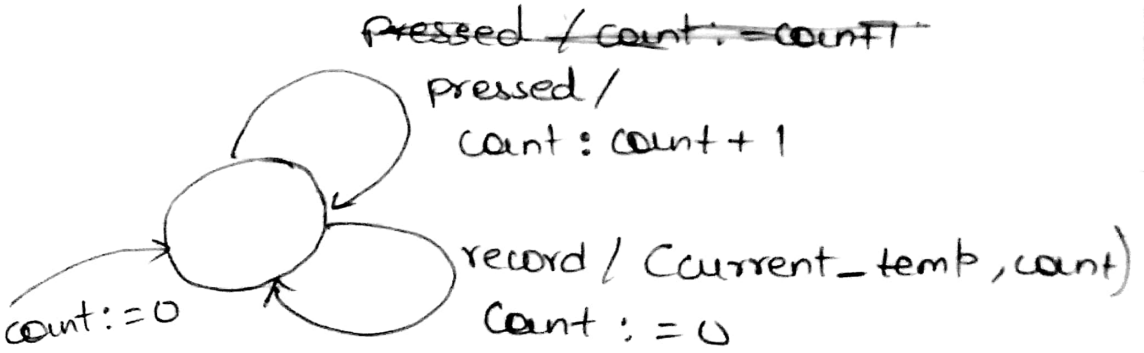
Name : KOTTE SAHITHI KRISHNA
Roll No: S20160010015
Question paper No : - 2

CPES End Exam

Input = { }
output : tick : pure
variable : count : {0, 1, ... 249}

The above FSM is an extended FSM, there are 250 variable values, if there is an simple FSM there need to be 250 states, here, an input is not required The output is tick, the $V_p \in \{present\} \cup \{absent\}$. According to the FSM, whenever count reaches 250, a tick is output and reset 0 because of count = count + 1

b) data logging system:-



input : { record, prssed } : pure
output : { ~~curtemp~~ current_temp , count }
variable : ~~count~~ . count $\in Z_+ \cup \{0\}$

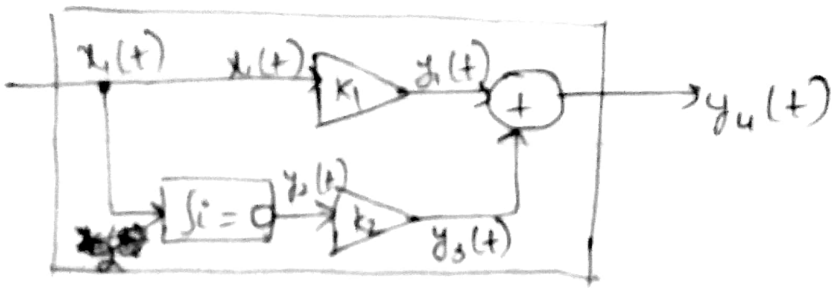
In the above FSM, the record input has a different FSM which counts till 30 sec and give {prssed} for log_count and current temperature. the outputs are curr-temp and count of the button was pressed and the count is set 0 by count : = 0, the count is positive integers with 0. This extended FSM has infinite no. of states equal to no of times the button was pressed.

Name: KOTTE SAHITHI KRISHNA

Roll No: S20160010045

CPES END EXAM Question paper NO:- 2

3) a)



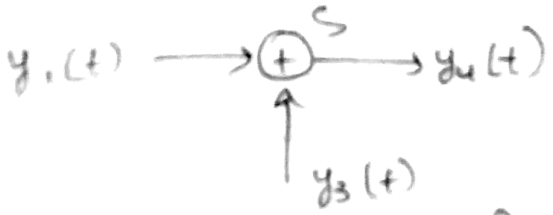
$x_1(t) : R \to R$

$\forall t \in R, \; y_1(t) = K_1 x_1(t)$

$\forall t \in R, \; y_2(t) = 0 + \int_0^t x_1(\tau) d\tau$

$\forall t \in R, \; y_3(t) = K_2 y_2(t)$

$= K_2 \int_0^t x_1(\tau) d\tau$



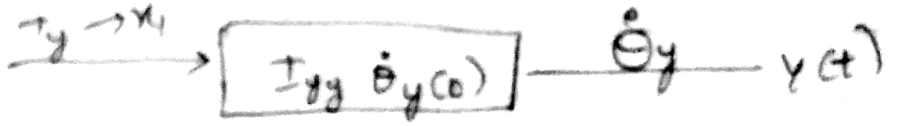
$S : (R \to R)^2 \to (R \to R)$

$\forall t \in R, \; y_4(t) = y_1(t) + y_3(t)$

$= K_1 x_1(t) + K_2 \int_0^t x_1(\tau) d\tau$

for $K_1 \quad (R \to R) \to (R \to R)$

$K_2 \quad (R \to R) \to (R \to R)$



the i/p is $x_1 = \tau_y$. $\theta_y(t) = \dfrac{\tau_{yy}}{I_{yy}}$ $K_1 = \dfrac{1}{I_{yy}\, \dot{\theta}_y(0)}$ and

Name:- KOTTE SAHITHI KRISHNA

Roll No:- SA16001004J

CPES END EXAM Question paper No - 2

b) Casual systems :- A system is casual if its output depend only on current and past inputs.

Casual:- $S: X \to Y \quad X = A^R \quad Y = B^R$.

$\forall\ x_1, x_2 \in X$ and $T \in R$

$$x_1|_{t \le T} = x_2|_{t \le T} \Rightarrow S(x_1)|_{t \le T} = S(x_2)|_{t \le R}$$

Strictly casual:- $\forall\ x_1, x_2 \in X$ and $T \in R$

$$x_1|_{t < T} = x_2|_{t < T} \Rightarrow S(x_1)|_{t \le T} = S(x_2)|_{t \le T}$$

The output at time t in strictly casual doesn't depend on input at time t. ~~It can be~~.

We know that adder is casual

Hence, the model consistes of integrator & adder.

∴ the model is also casual.