Srushti Shah
19107l902

EXPERIMENT : No.

Parallel Computing - MST

PAGE No.
DATE

1) The parallel processing structure are:-

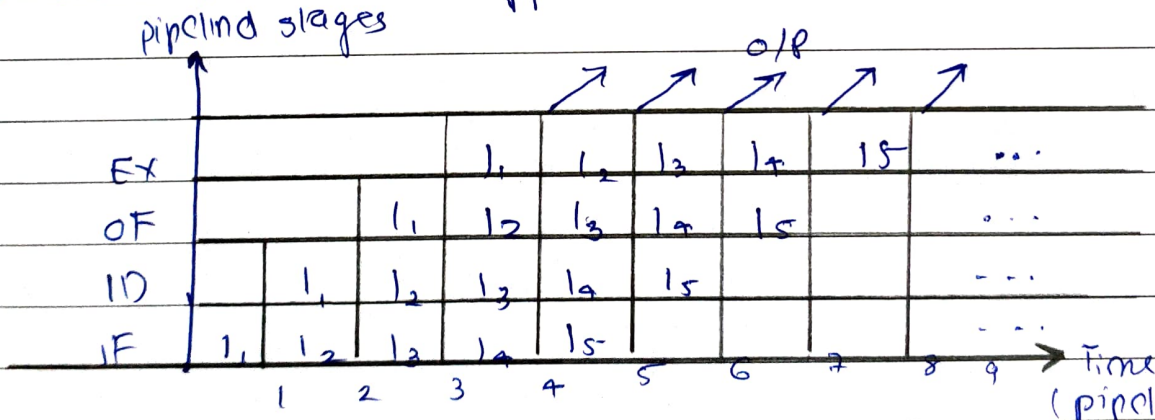
(i) Pipeline computer

→ It performs overlapped computations to exploit temporal parallelism.

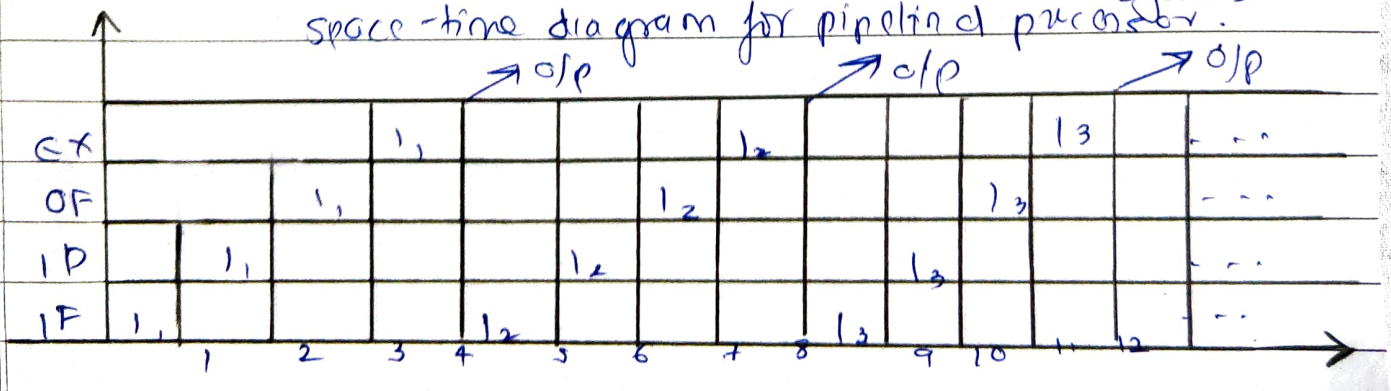
→ The process of executing an instruction in computer involves
(i) instruction fetch (IF) from main memory
(ii) OF (operand fetch)
(iii) ID (Instruction Decoding), identifying the operation to be performed
(iv) EX (execution) of decoded arithmetic logic operation

→ In a non-pipelined computer the 4 steps must be completed before the next instruction

→ In a pipelined computer successive instructions are executed in an overlapped fashion.



space-time diagram for pipelined processor.

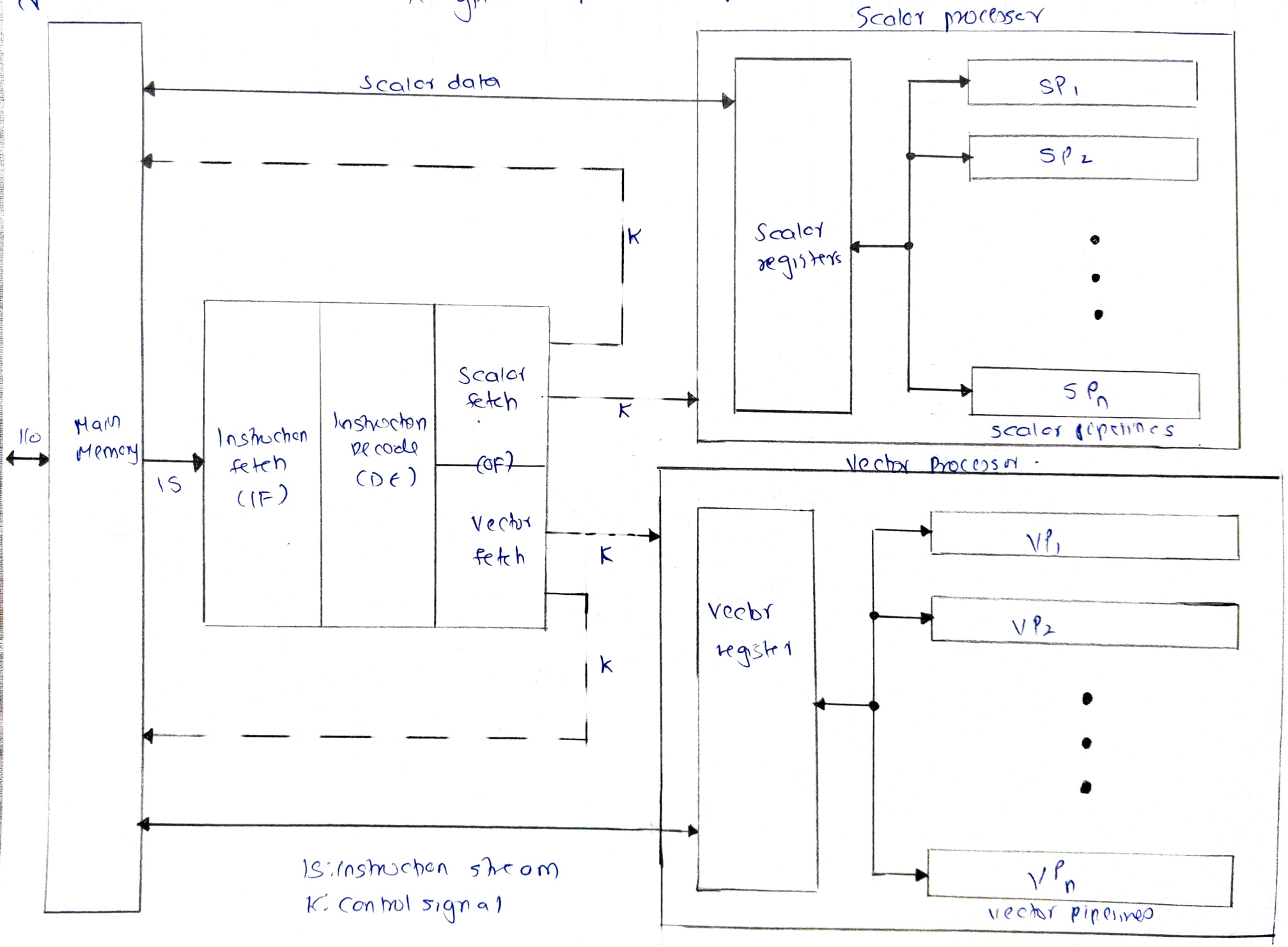


space-time diagram for non-pipelined processor.

Sundaram

Teacher's Sign :

Functional structure: A typical pipeline computer



EXPERIMENT : No.

PAGE No.
DATE

- An instruction cycle consists of multiple pipeline cycles.
- A pipecycle can be set equal to delay of slowest stage
- The flow of data from stage to stage is triggered by a common clock of pipeline (synchronized operations)
- Interface latches are used between adjacent segments to hold the intermediate result.

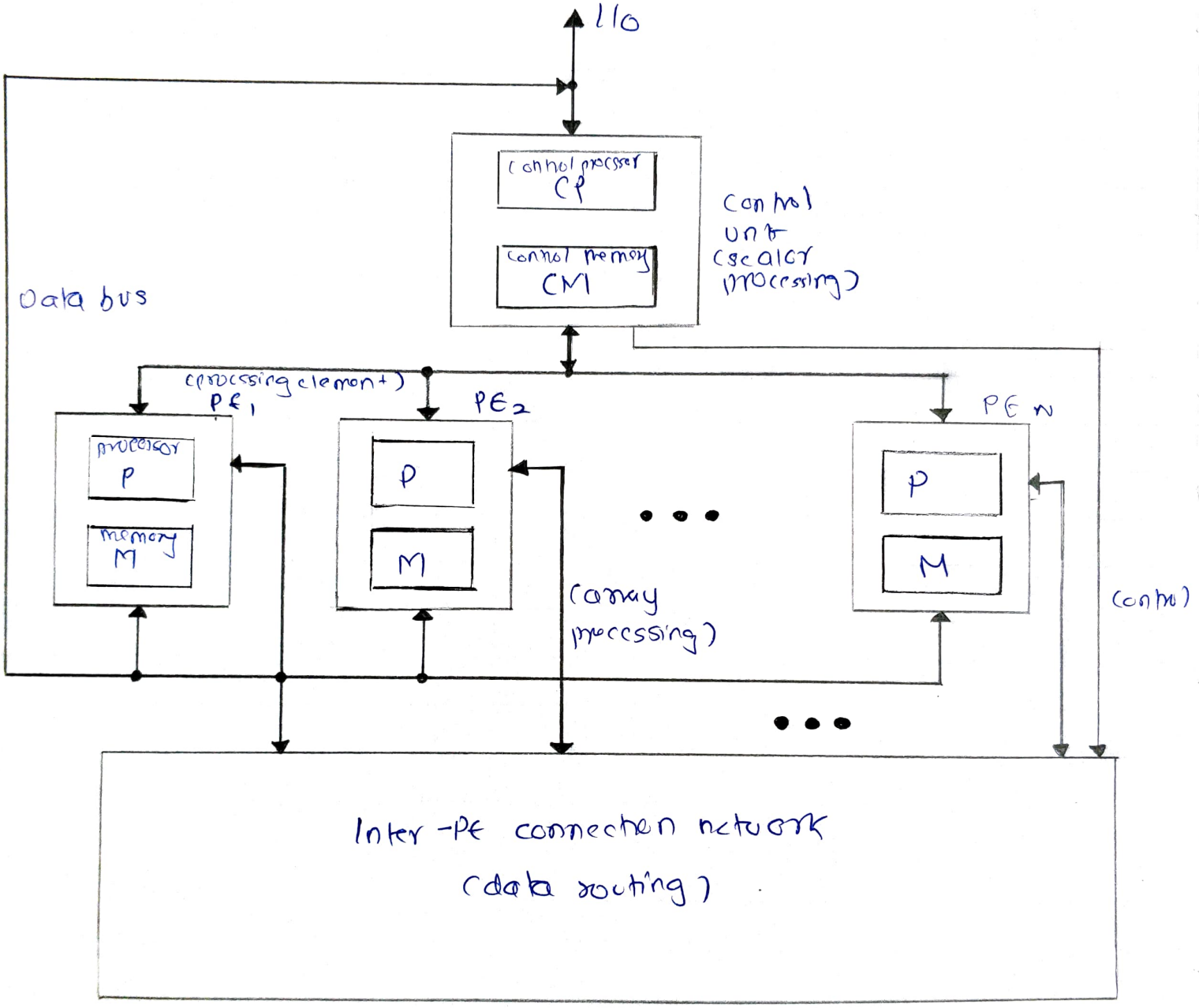
- The k-staged pipelined processor could be at most k times faster, but due to memory conflicts, data dependency branch & interrupts, this ideal speed up may not be achieved for out-of sequence computations
- For some CPU-bound instructions, the execution phase can be further partitioned into multiple-stage arithmetic logic pipeline.
- Due to overlapped instruction & arithmetic operations, pipelined machines are better tuned to perform some operations repeatedly.
- → Few issues with the pipelined computer designed include job sequencing, collision prevention, congestion control, reconfiguration and hazard resolution.

(ii) Array Computers.

- An array processor is a synchronous parallel computer with multiple arithmetic logic units called processing elements (PE) that can operate in parallel in a lock-step fashion.
- By replication of ALU, one can achieve the spatial parallelism
- The PE's are synchronized to perform the same functions at same time. An appropriate data-routing mechanism must be established among PE's

Sundaram

Teacher's Sign :



A typical Array Computer (SIMD) function structure.

EXPERIMENT : No.

PAGE No.
DATE

In a typical array computer

- Scalar & control type instructions are directly executed in CU (control unit)
- Each PE has ALU and register with local memory.
- PEs are interconnected by data-routing network.
- The interconnection pattern to be established for specific computation is under program control from CU
- Vector instructions are broadcast to PE's for distributed execution over different computer operands fetched directly from local memory.
- IF & decode is done by CU
- The PEs are passive devices without instruction decoding capabilities.

2) The paged memory system:
- The virtual space is partitioned into pages, which can be readent in matching size blocks called page frames in memory
- Each virtual address contains
  (i) a virtual page number ip which is the mapped field
  (ii) displacement iw of the word within the page, which is the unmapped field.
- The address map consists of a page table (PT) from which is read the corresponding base address of page frame if the page exists in main memory.
- simplest page table contains one entry for each possible virtual page.
- There is one PT for each process & PT is created in main memory at initiation of process.
- A PTBR (page table base register) in each process contains the base addr of page table of process that is currently running on that processor
- The PT entry must be accessed by indexing into page table array.

→ For each page table entry (PTE) consists of a valid bit (F), a permissible access code (RWX), a memory bit (M) and page frame address (PFA)
- The valid bit if set indicates that the page exists, or is not null. A page that is null will have to be created when referenced.
- A page is said to be active with respect to a process if it is resident in main memory
- The M flag is set in PTE of that process & PFA field contains the add' of page in memory.

Sundaram

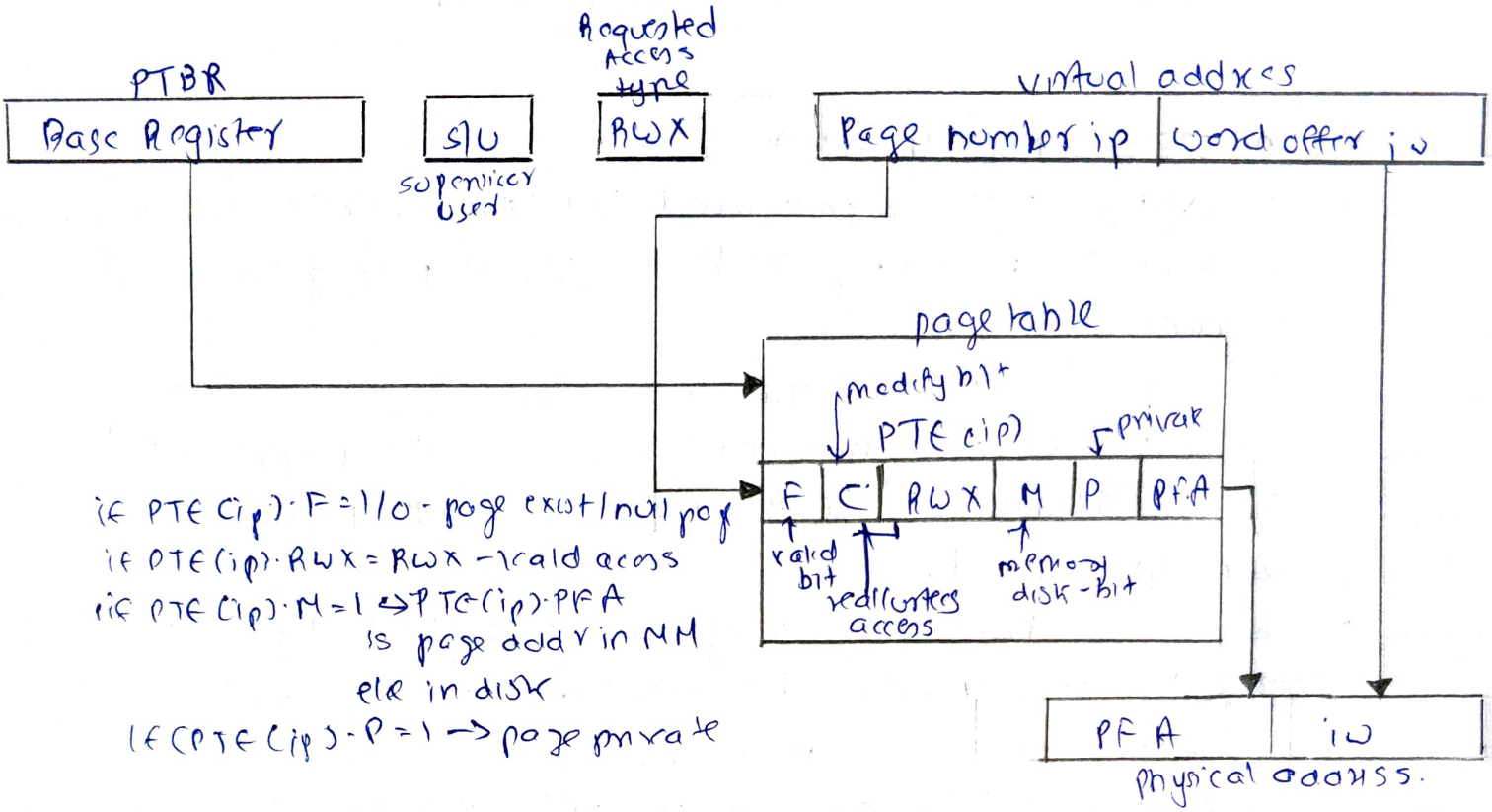
Teacher's Sign.:

- In constrast a nonnull page is inactive, virt to a process & M is cleared, then PFA contains disk address of page.

→ Mapping of virtual to physical address.

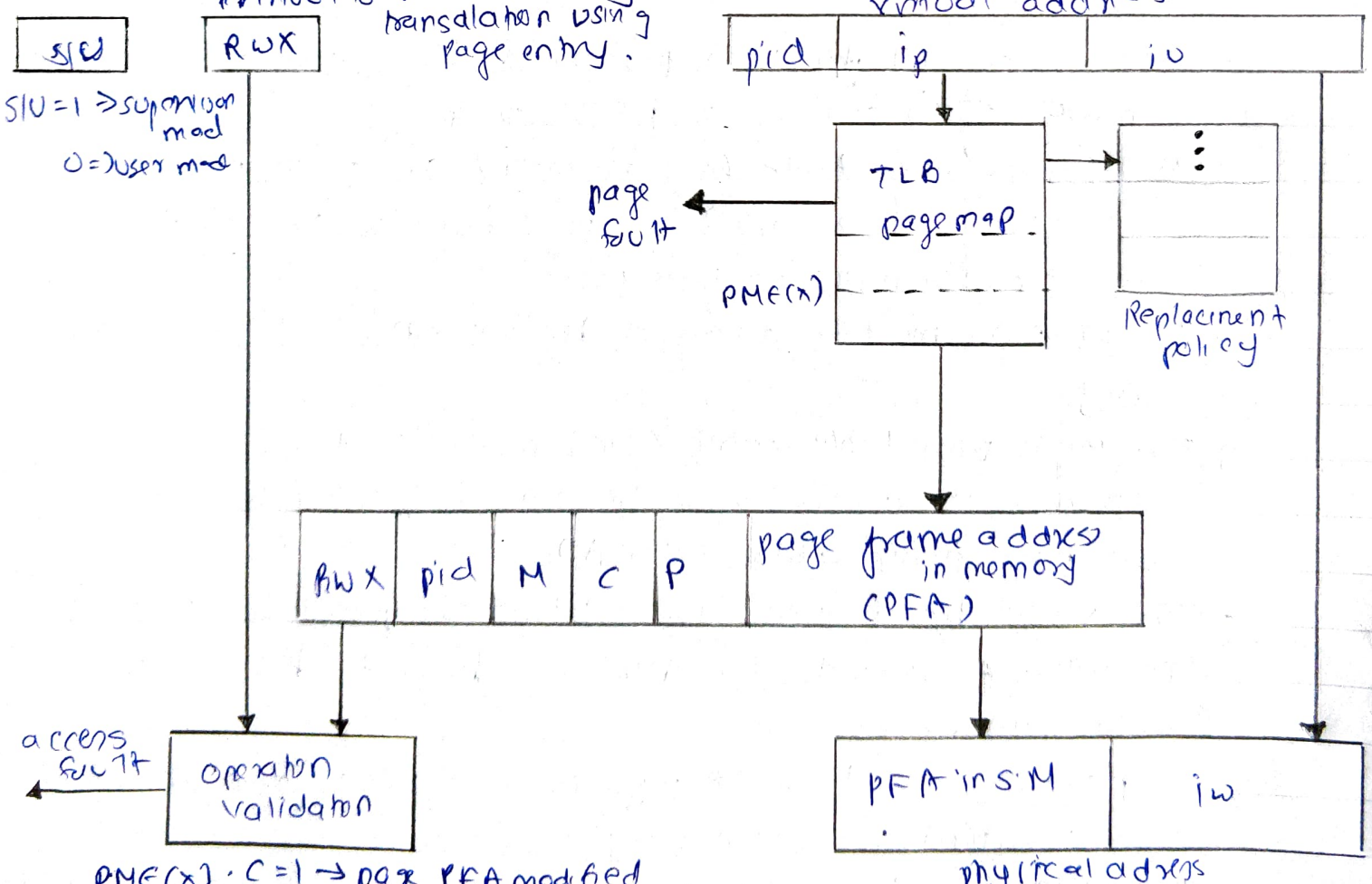
- A virtual address generated by a running process, the virtual-to-real address translation involves associative comparision of virtual page number in with all PME (page memory entries that contain the same process identification as current running process
- If there is a match, the page-frame number is retrieved and physical address formed by concatenating the displacement with PFN.
- If there is no match, a page fault interrupt occurs, which is serviced to locate the page.
- Moreover, if a page access key presented by virtual addr is ~~number the PID, an access~~ does not match the RWX field of PME with correspond virtual page number and PID, an access violation is trapped.
- When a reference page is modified, a modified bit (d) of PME is set in page map. by replacement or memory update policies.
- When a page fault occurs because the virtual page number in was not found in TLB, a dynamic addr translation is requested, using page table which is resident in main memory
- The virtual page number in is used as an index, is added to the page table addr in PTBR and resulting addr is used to access the PTE.

Virtual to Real page address translation.



if PTE(ip)·F = 1/0 - page exist/null page
if PTE(ip)·RWX = RWX - valid access
if PTE(ip)·M = 1 ⇔ PTE(ip)·PFA is page addr in MM else in disk
IF (PTE(ip)·P = 1 → page private

Virtual to real addressing translation using page entry.



S|U = 1 ⇒ supervisor mod
0 ⇒ user mod

PME(x)·C = 1 → page PFA modified.
PME(x)·P = 1 → page is private
PME(x)·pid is process identificator number
PME(x)·PFA is page frame address.

EXPERIMENT : No.

PAGE No.
DATE

- If the PTE indicates that the page in not in main memory, the running process is blocked or suspended.
- A context switch is made to another ready to run process, while the page is transferred from disk to main memory. & PTE is updated.
- If page is in memory, the TLB is updated with virtual page number & page's page addr pair.

Disadvantages :- of Memory Mapped page system :-

(i) It is inefficient as it requires two memory accesses for each data accessed.

(ii) For paged memory system can become inefficient if virtual space is large. The size of page table can become unreasonably large. Example: for a 32-bit virtual address and a 1024 (1K) byte page size. the page addr field requires 22 bits thus 2^{22} PTE

(iii) There is no mechanism for a reasonable implementation of sharing. The size of a program space is not always an integral no. of pages, hence internal fragmentation occurs in memory because the last part of last page is wasted

(iv) There can also be table fragmentation because of physical memory are occupied by page tables and so are unavailable to virtual pages.

Sundaram

Teacher's Sign.: