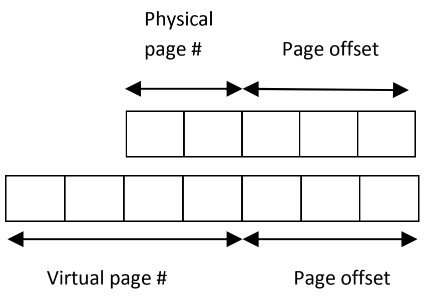
Inverted Page Table Implementation Example:
Page Size: 8 bytes – (3 address bits)
No. of Physical Pages: 4 pages (2 page bits)
Physical Address: 5 bit addressing
No. of Virtual Pages: 16 pages (4 page bits)
Virtual Address: 7 bits

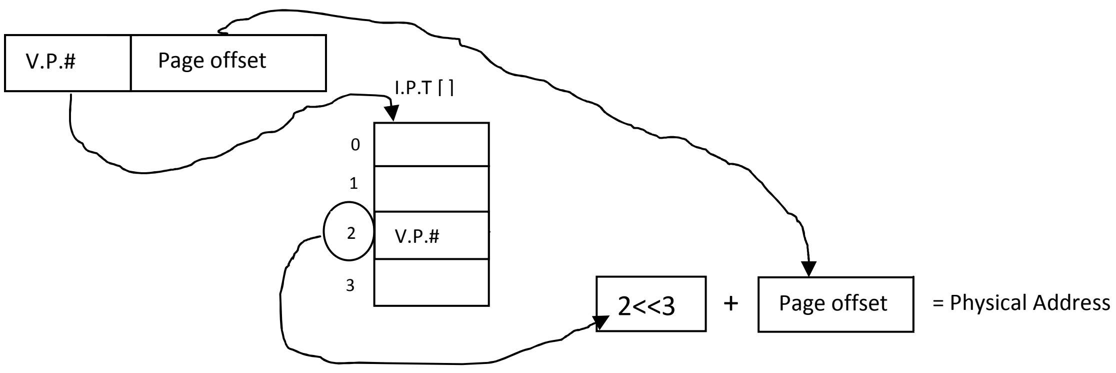


Inverted Page Table entries: 4

```
typedef  struct InvPageTable
{     pid;    /*process id, -1 indicate physical page available*/
      vp;     /* virtual page number */
}
InvPageTable      IPT[4];
```

Virtual address to Physical address Translation

1. CPU issues a Virtual Address
2. Extract the virtual page number (evp) from the given virtual address
3. Go thru IPT[ ].vp to find the match
4. if IPT[i].vp == evp, phys_address = (i<<2) + page_offset portion of the virtual address;
   else it is page fault. Load the needed page



Loading a page of process pnum

1. Go thru IPT[] to find available page.
   If (all busy) then use LRU to select one to be evicted
2. Set IPT
   IPT[available].pid = pnum;
   IPT[available].vp = virtual_page portion of virtual_address;
3. Start loading the page
   Loading start address = (available<<3) + 0;