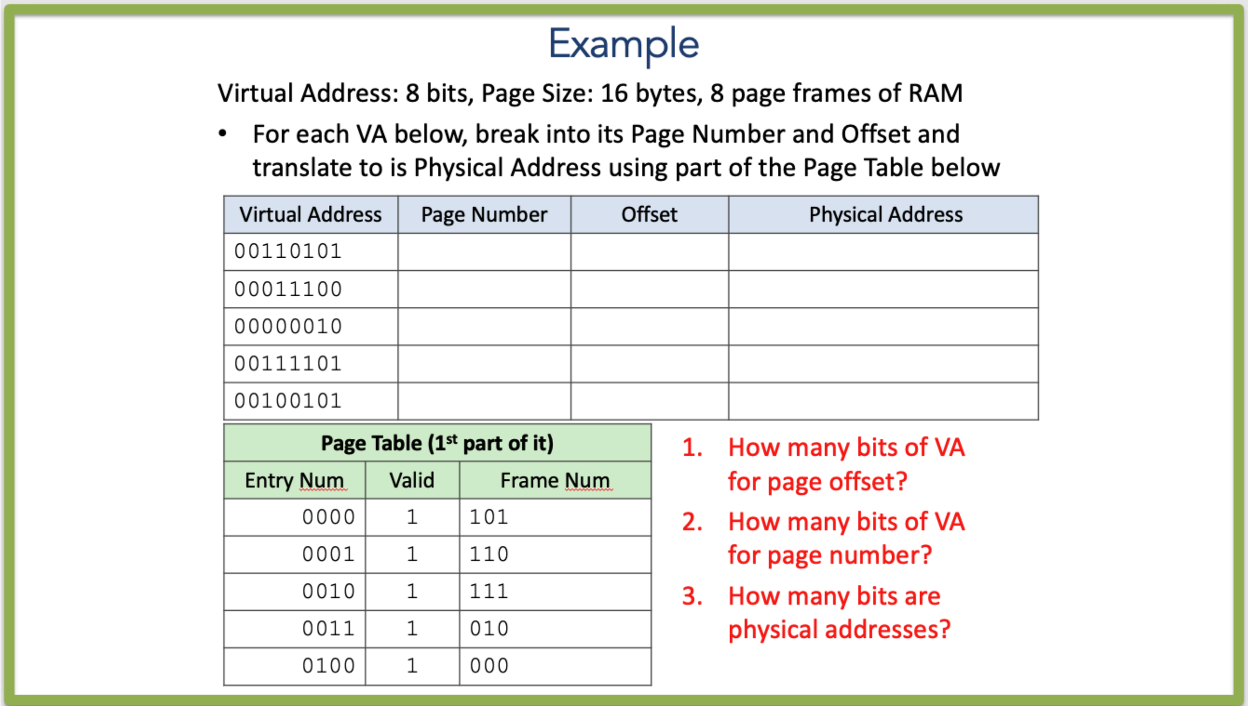


Example

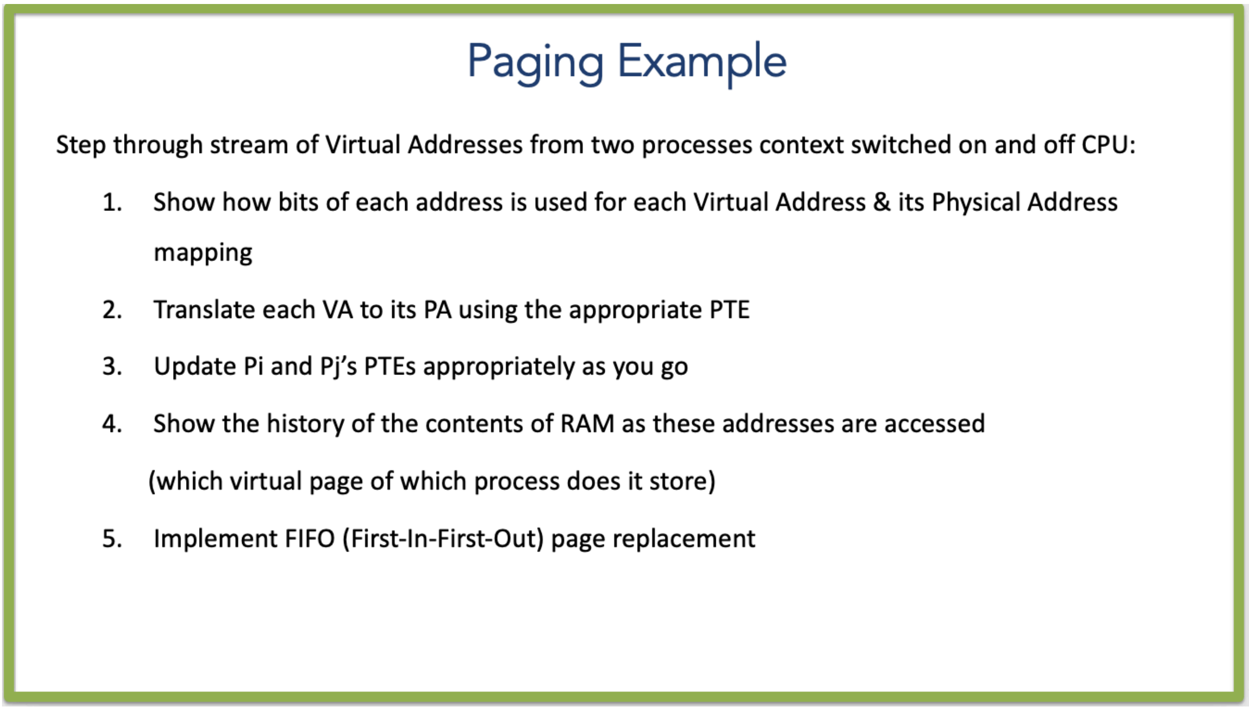
Virtual Address: 8 bits, Page Size: 16 bytes, 8 page frames of RAM

- For each VA below, break into its Page Number and Offset and translate to is Physical Address using part of the Page Table below

Virtual Address	Page Number	Offset	Physical Address
00110101			
00011100			
00000010			
00111101			
00100101			

Page Table (1st part of it)		
Entry Num	Valid	Frame Num
0000	1	101
0001	1	110
0010	1	111
0011	1	010
0100	1	000

1. How many bits of VA for page offset?
2. How many bits of VA for page number?
3. How many bits are physical addresses?

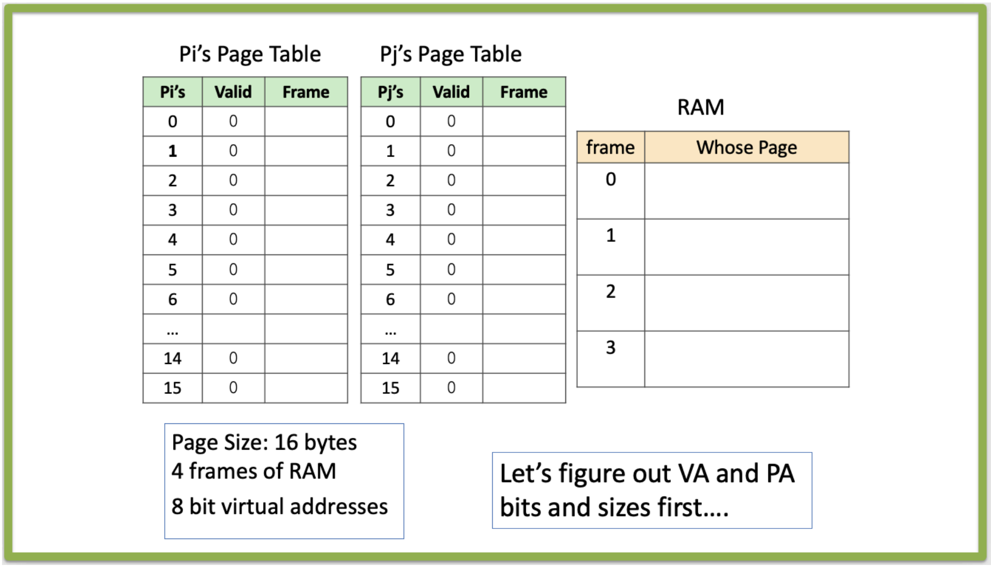


Paging Example

Step through stream of Virtual Addresses from two processes context switched on and off CPU:

1. Show how bits of each address is used for each Virtual Address & its Physical Address mapping
2. Translate each VA to its PA using the appropriate PTE
3. Update Pi and Pj's PTEs appropriately as you go
4. Show the history of the contents of RAM as these addresses are accessed

   (which virtual page of which process does it store)
5. Implement FIFO (First-In-First-Out) page replacement



Pi’s Page Table

Pi’s	Valid	Frame
0	0	
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
…		
14	0	
15	0	

Pj’s Page Table

Pj’s	Valid	Frame
0	0	
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
…		
14	0	
15	0	

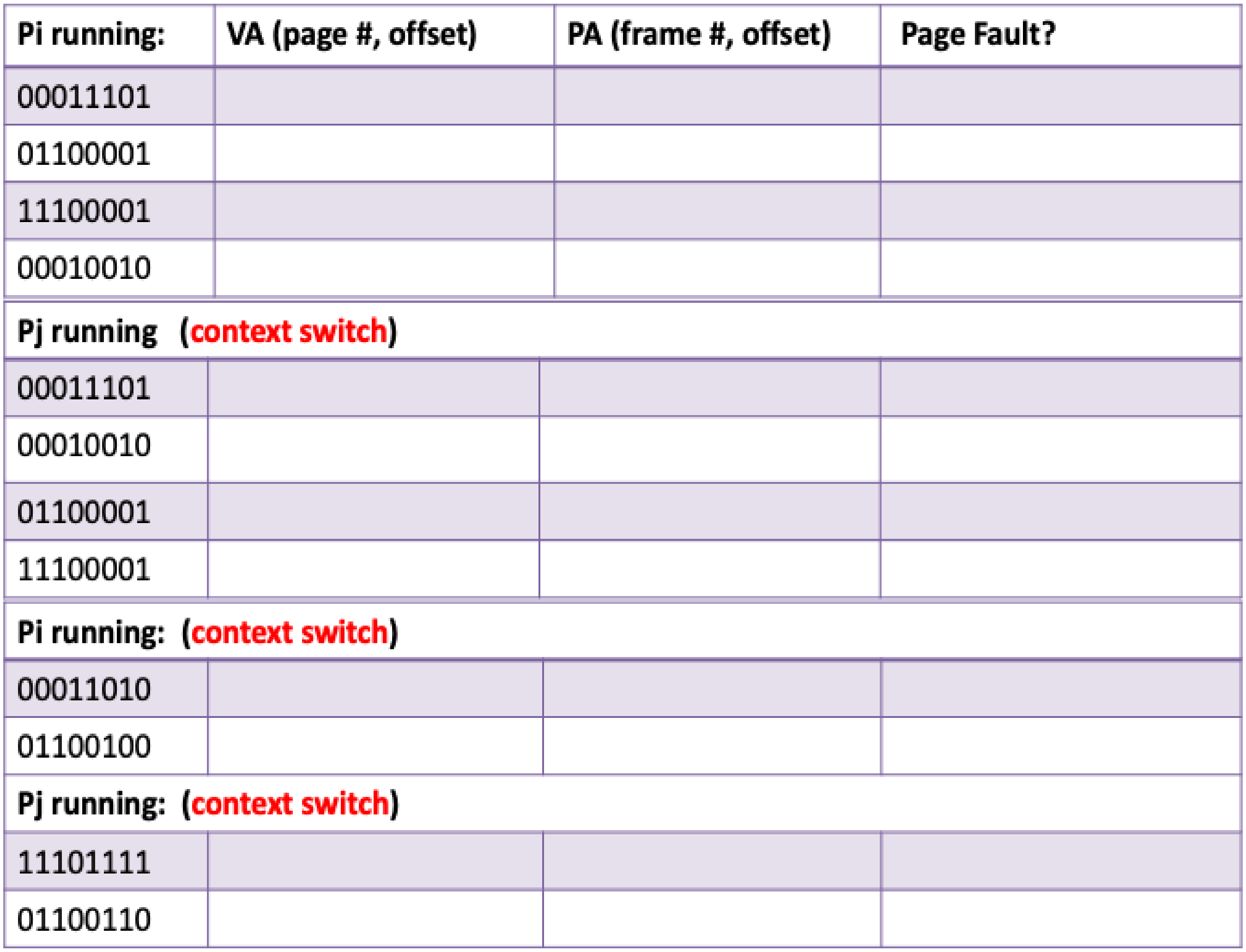
RAM

frame	Whose Page
0	
1	
2	
3	

Page Size: 16 bytes
4 frames of RAM
8 bit virtual addresses

Let’s figure out VA and PA bits and sizes first….

order of memory accesses of 2 processes, Pi and Pj, running on CPU (note context switches)



Pi running:	VA (page #, offset)	PA (frame #, offset)	Page Fault?
00011101			
01100001			
11100001			
00010010			
Pj running (context switch)			
00011101			
00010010			
01100001			
11100001			
Pi running: (context switch)			
00011010			
01100100			
Pj running: (context switch)			
11101111			
01100110			