

Lec 16a.

int TLB_replacement_LRU (page-t p-num, ⁴⁻¹/* new page-num,
frame-t f-num, /* new frame-num,
tlb-t *tlb).

or
int TLB_replacement.LRU(tlb_entry-t new_entry, tlb-t *tlb)

```
typedef struct {
    page-t page-num;
    frame-t frame-num;
    bool valid;
    age-t age;
} tlb_entry-t;
```

	page-num	frame-num	valid	age
0			*	
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

```
typedef struct {
    tlb_entry-t tlb_entry[TLB_SIZE];
    u-int-t next_tlb_ptr;
} tlb-t;
```

4-2

oldest_age = 0;

oldest_ptr = 0;

found = FALSE;

```

=> for (i = 0; i < tlb->next_tlb_ptr; i++)
    assert(tlb->tlb_entry[i].valid == TRUE;

```

```

/* found : update age */

```

```

=> if (tlb->tlb_entry[i].page_num == p_num) {
    assert(tlb->tlb_entry[i].frame_num == f_num);

```

```

    found = TRUE;

```

```

    tlb->tlb_entry[i].age = 0;

```

this is
an error.
Do NOT
use break
here!

```

    break; /* not recommended : use while rather than
    for */

```

```

/* not found : update age, locate the least recently used one

```

```

=> else {

```

```

    tlb->tlb_entry[i].age++;

```

```

    if (tlb->tlb_entry[i].age > oldest_age) {

```

```

        oldest_age = tlb->tlb_entry[i].age;

```

```

        oldest_ptr = i;

```

```

    }

```

```

} /* end else */

```

```

} /* end for */

```

part 1

① Search TLB

② locate the least recently used one (oldest)

4-3
/* cont. after for */
/* what should we do if found? Nothing */

if (found == FALSE) {

/* There is a free tlb entry. How do we know? */

if (tlb->tlb_entry[next_tlb_ptr].valid == FALSE) {

/* update the new entry */

tlb->tlb_entry[tlb->next_tlb_ptr].page-num = p-num;

✓ .frame-num = f-num;

assert(tlb->next_tlb_ptr >= 0
tlb->next_tlb_ptr < TLB_SIZE);

✓ .valid = TRUE;

✓ .age = 0;

/* update next_tlb_ptr */

tlb->next_tlb_ptr++;

}

else { /* No free entry; oldest one is replaced */

tlb->tlb_entry[oldest_ptr].page-num = p-num;

✓ .frame-num = f-num;

✓ .age = 0;

assert(

✓

.valid == TRUE);

} /* end else */

} /* end if not found */

part 2 ① replace the oldest entry

New Design: Integrate FIFO and LRU 4-4
TLB-replacement (p-num, f-num, ~~x~~tlb)

```
if (there is a free tlb entry) {  
    add the new tlb entry;  
    update next-tlb-ptr;  
}
```

```
}
```

```
else {  
    switch (replacement-policy) {
```

```
        case FIFO:   
            TLB_replacement_FIFO(p-num, f-num, tlb);  
            break;
```

```
        case LRU:   
            TLB_replacement_LRU(p-num, f-num, tlb);  
            break;
```

```
        default:   
            printf("Error, this replacement policy hasn't been  
                implemented!\n");
```

```
    }
```

```
}
```