

# Efficient Translations

⦿ **Problem** : expensive memory access

- One-page table : one table lookup + one fetch
- Two-page table (32 bits) : 2 table lookups + one fetch
- 4-page table (64 bits) : 4 table lookup + one fetch

✓ **Solution : Translation Lookaside Buffer (TLB)**  
cache translations in hardware to reduce lookup cost

# Translation Lookaside Buffers (TLBs)

## **Translation Lookaside Buffers**

special hardware to translate virtual page #s into PTEs  
(not physical address) in a single machine cycle

- Typically 4-way to fully associative cache (all entries looked up in parallel)
  - Cache 32-128 PTE values (128-512K memory)
- ➔ TLBs exploit locality : processes only use a handful of pages at a time  
TLB hit rate is a very important for performances  
(>99% of translations)