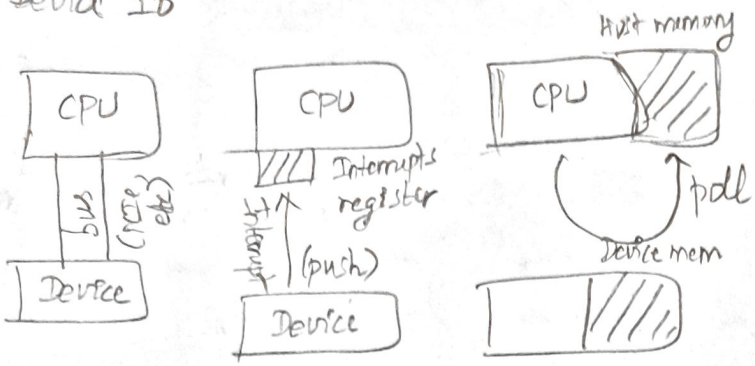
① Device IO

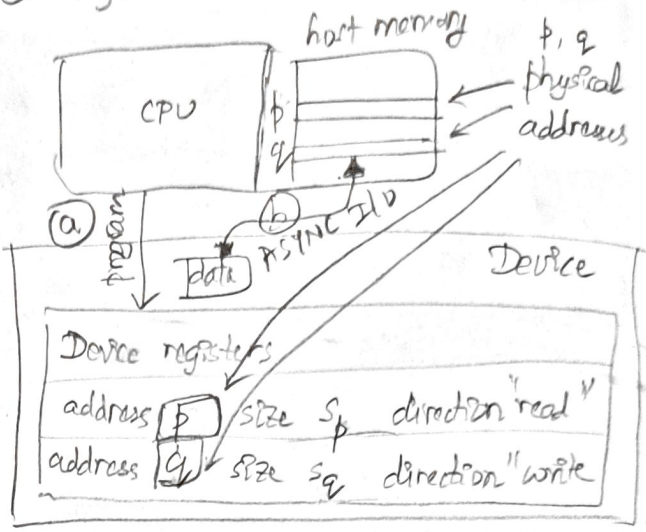


ⓐ Mix Interrupts with polling

ⓑ network: interrupt → interrupt disable
→ poll for timeout / until no data
→ reenable interrupts

② Direct Memory Access (DMA)

use CPU cycles when data moved
① large data ② scattered data

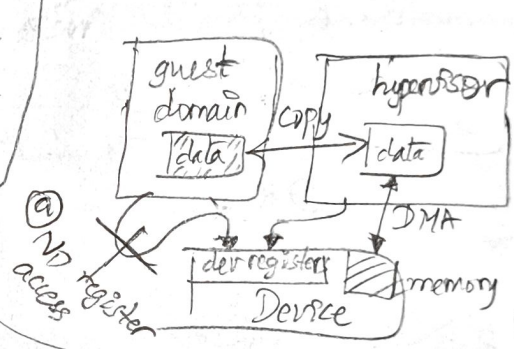


③ how should IO be virtualized?

ⓐ guest can't (directly) program sensitive device registers

ⓑ device I/O on behalf of guest must be vetted by the hypervisor — addresses part of guest domain.

④ Approach 1 data copy

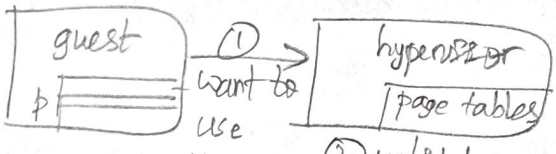


ⓑ DMA to/from trusted memory programmed by dom 0

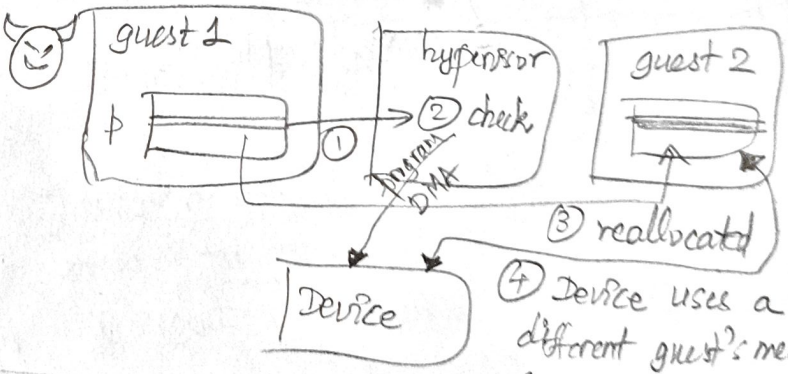
ⓒ data copy to/from guest

⑤ Approach 2 Validation of guest memory in DMA descriptors

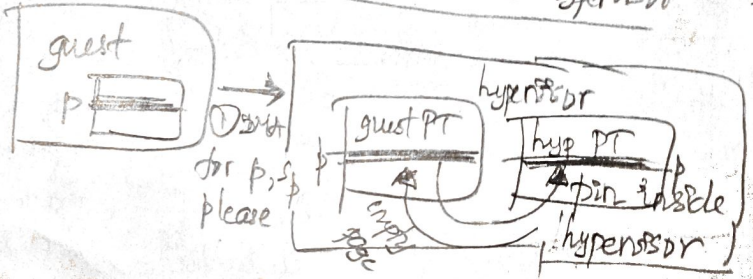
ⓐ Don't copy. Validate guest memory



ⓑ problem: time of check to time of use



⑥ Approach 3. Pin pages inside hypervisor



page = pinning

ⓐ upon DMA request validation, pin page inside hypervisor's own page tables. Cannot be reallocated to anyone else until DMA complete.

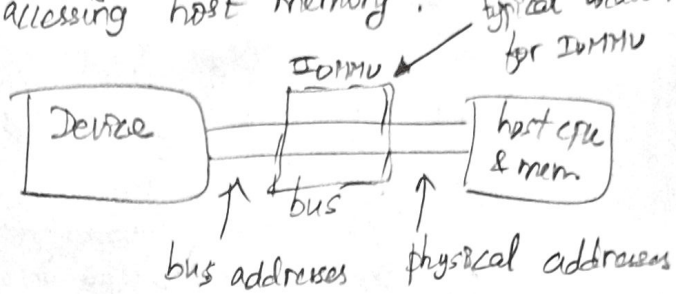
(7) Need for hardware support

→ validation per DMA data transfer

→ 30% of native network throughput

(8) IO Memory Management unit (IOMMU)

Hardware to translate & isolate devices accessing host memory.



(8a) translation: useful when device can only access part of host memory (e.g.: 32 bit device versus 64 bit CPU)

(8b) Isolation IOMMU has a page table

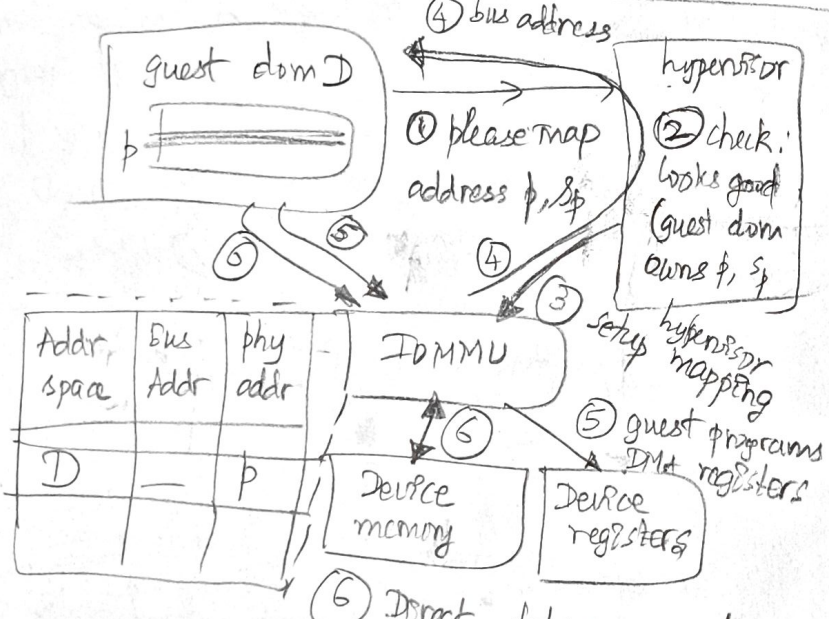
Entries are only mapped/unmapped by privileged entity (hypervisor) after validation.

(8c) page table structure

address space ID	bus address	physical address

Key → value

(9) Data movement with IOMMU & address space isolation



⑥ Direct data movement to/from guest memory checked by IOMMU

ⓐ separate address spaces per guest domain

ⓑ validation at map/unmap time; guest gets a bus address

ⓒ direct DMA access from guest for all mapped bus addresses (checking directly in hardware IOMMU at the time of use)

ⓓ share most other resources

(10) Single Root IO virtualization (SR-IOV)

ⓐ expose single device as multiple devices (PF/VF/UNIC)

ⓑ dedicated DMA data movement registers per VF

ⓒ sensitive registers programmed by privileged entity