Computersystemen 2 Theorie

3. I/O Beheer

Karel de Grote Hogeschool

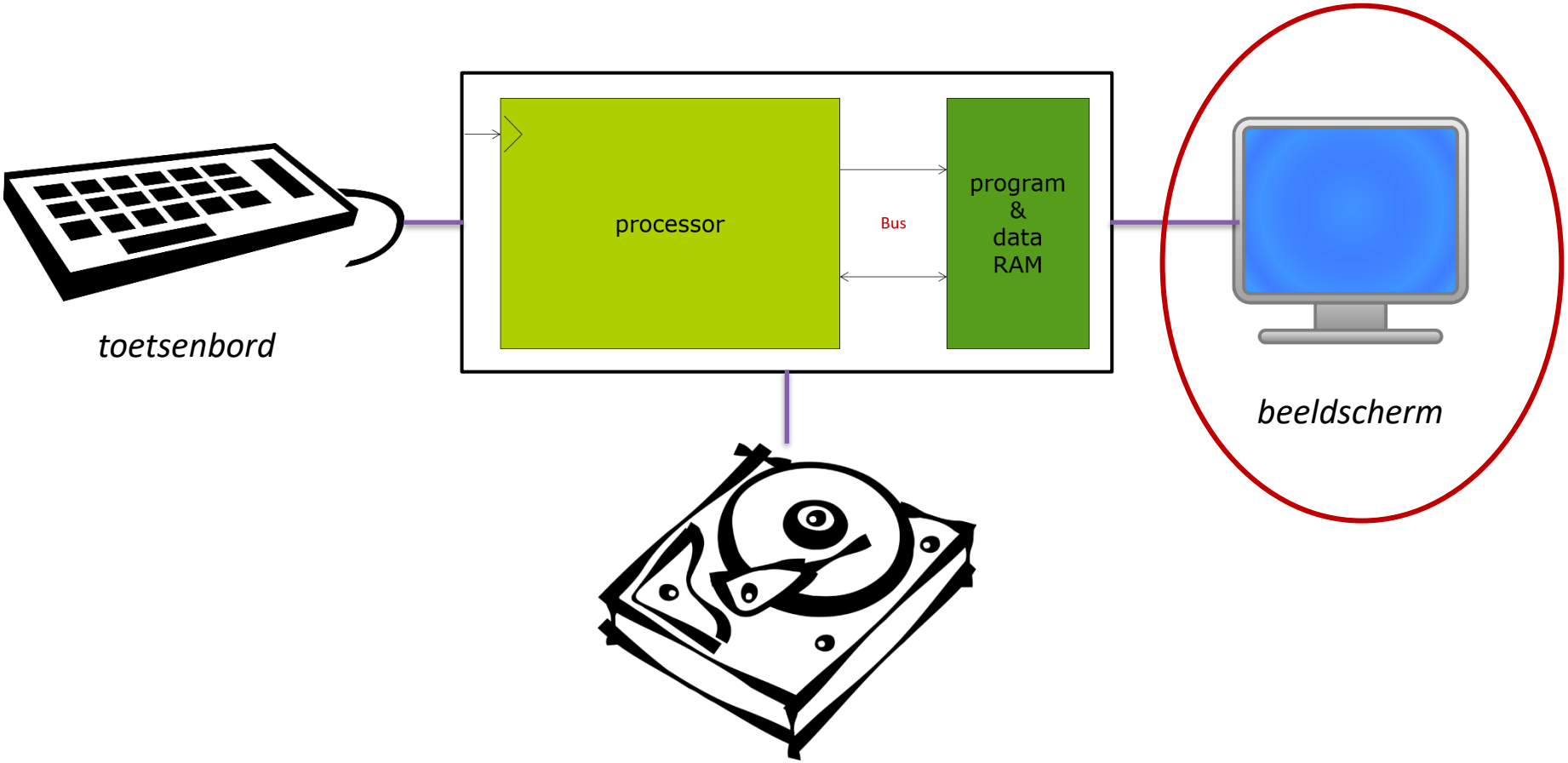
Input/Output

- OK, computer start op
- maar hoe kan je iets op een scherm zetten?
- hoe weet je welke toets een gebruiker heeft aangeslagen?
- hoe lees je iets van de harddisk drive (HDD)/solid state drive (SSD)?
- De BIOS biedt wel iets aan, maar veel te beperkt

Inhoud

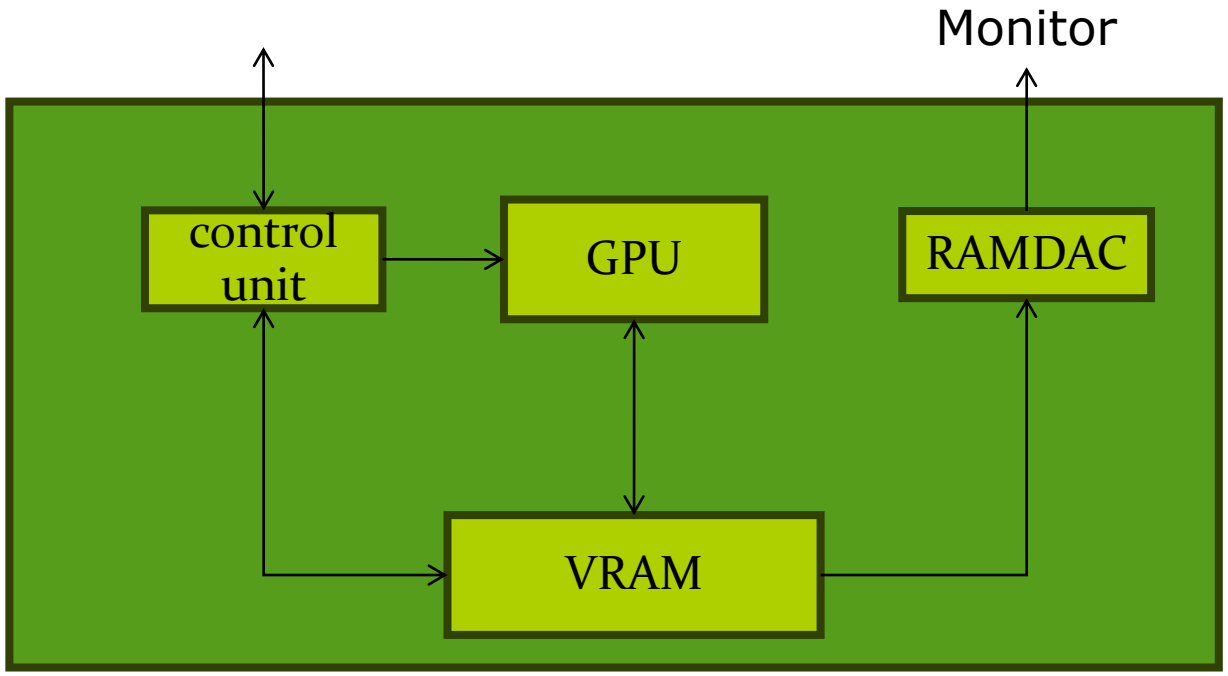
- Beeldscherm
- Toetsenbord
- Disk drive
- Herhalingsvragen

Input/Output - Beeldscherm



Grafische kaart

- Het beeldscherm wordt bestuurd door grafische kaart

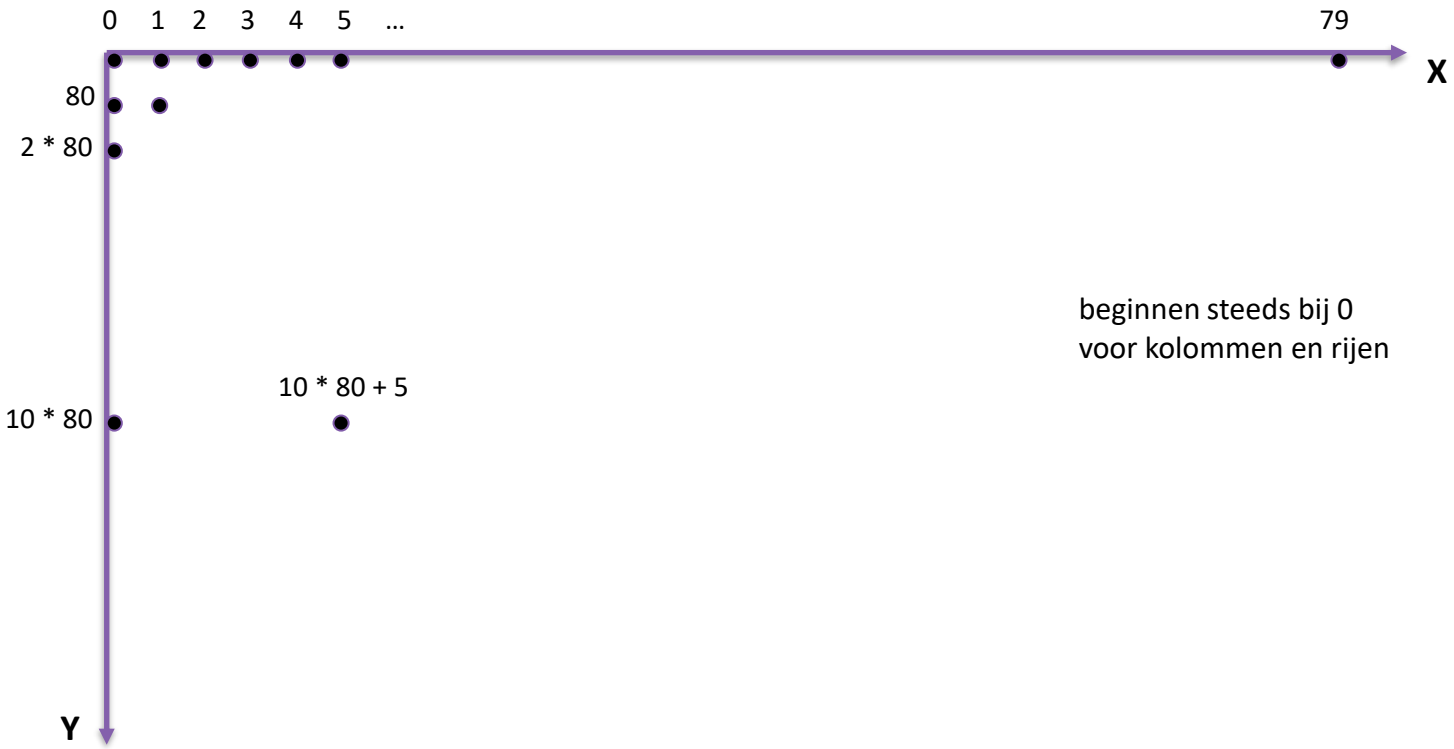


Video modes

- video geheugen bevat data voor beeld
- text-mode
  - 80x25 karakters
  - 1 byte per karakter op het scherm
- grafische mode
  - pixels van links naar rechts en van boven naar onder
  - indexed
    - 1 byte per pixel
    - byte is een kleurnummer uit tabel
    - tabel bevat eigenlijke kleuren
  - true-color
    - 3 bytes per pixel (RGB)

Video RAM

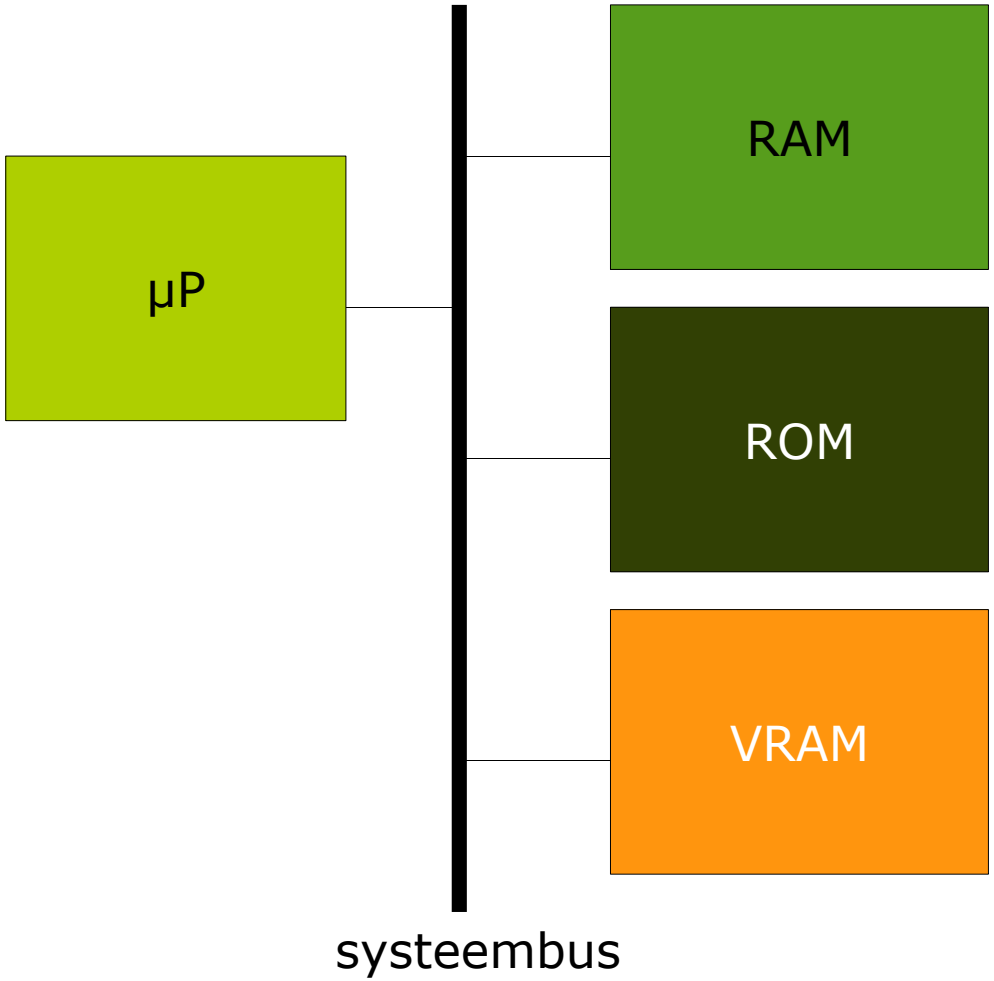
- voorbeeld 1: text-mode
  - zet karakter 'A' op kolom 5 van regel 10
  - VRAM[10*80+5] = 65



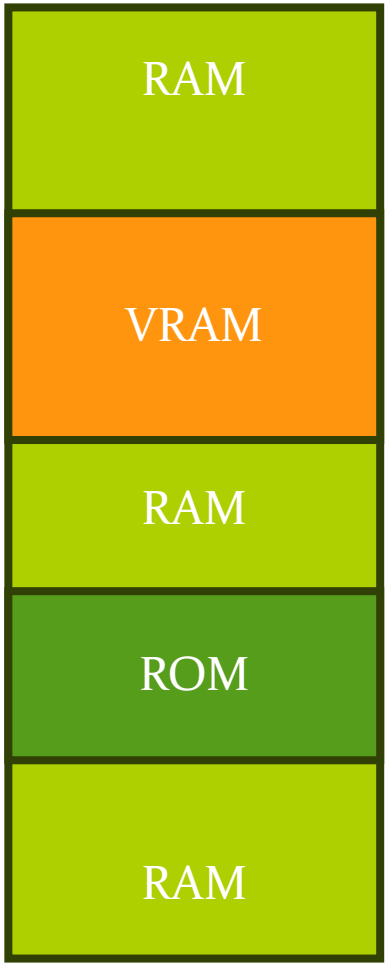
Memory-mapped I/O

- Hoe kan de processor nu een karakter/pixel tekenen op het scherm?
- hoe krijgt de processor toegang tot de VRAM?
- --> memory-mapped I/O

VRAM – Memory-mapped



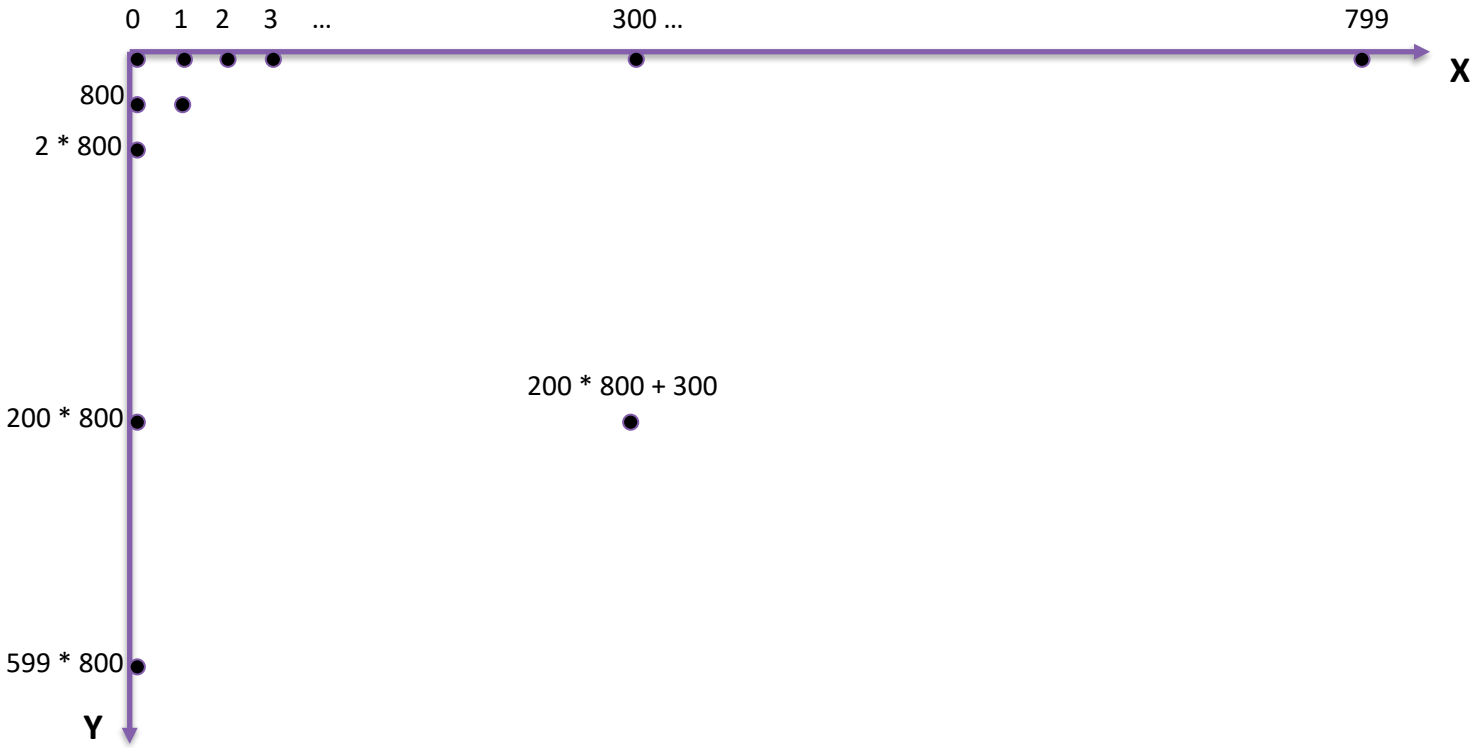
VRAM – Memory-mapped



VRAM – Memory-mapped

- voorbeeld 1:
  - stel dat VRAM gemapt is op 0xA0000
  - video kaart staat in text-mode
  - zet karakter 'A' op kolom 5 van regel 10
  - VRAM[805] = 65 => RAM[0xA0325]= 65
- voorbeeld 3 (oefening):
  - VRAM gemapt op 0x1E2345
  - video kaar staat in indexed graphical mode (800x600)
  - zet pixel (300, 200) op kleur nummer 5
  - geef adres hexadecimaal

Beeldscherm



- VRAM[160300]=VRAM[0x2722C]=5
- RAM[0x1E2345 + 0x2722C]=RAM[0x209571]=5

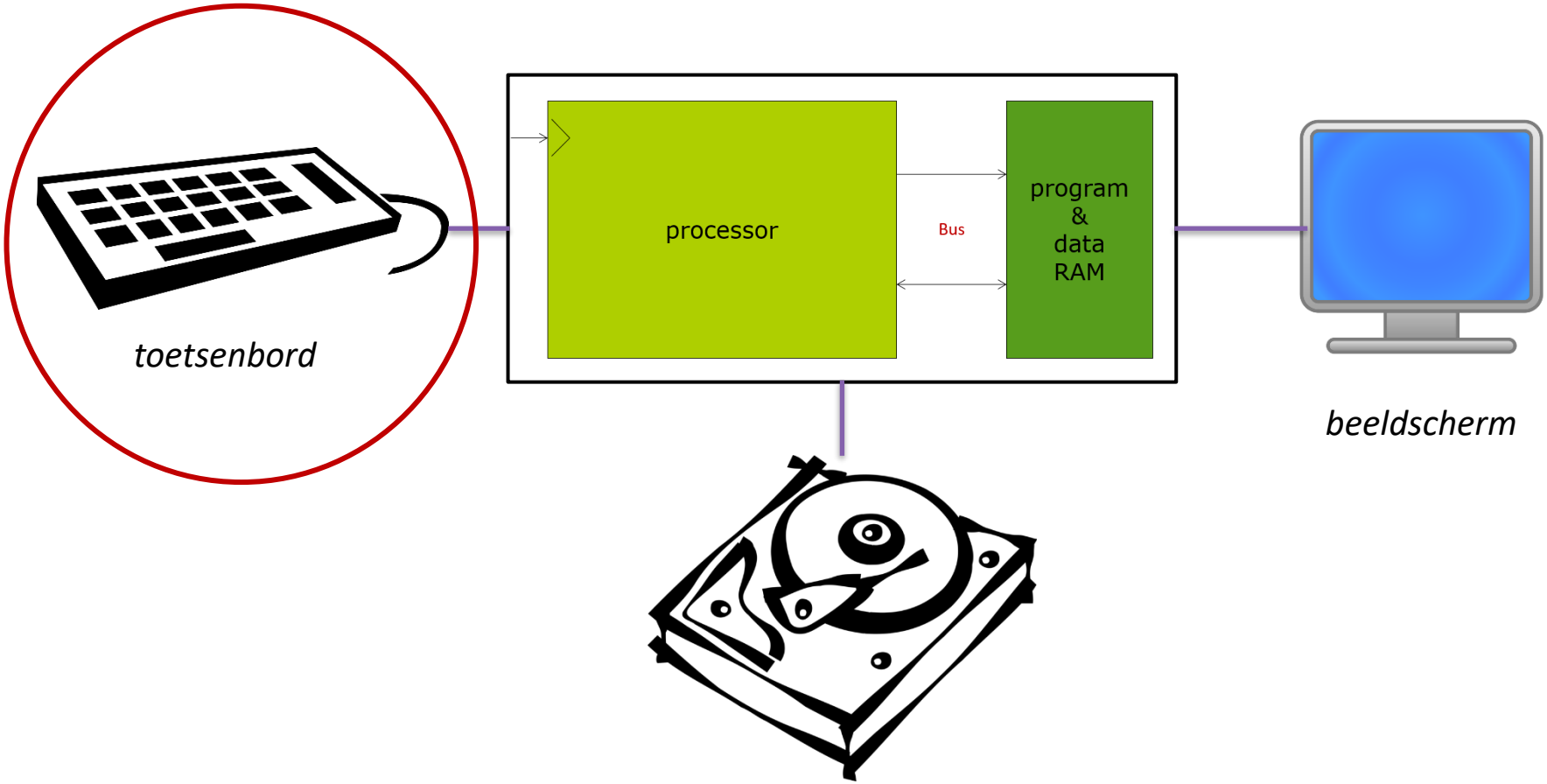
Beeldscherm

- sommige operaties kosten veel tijd
  - bv: drawCircle()
  - ==> processor moet alle pixels berekenen
- oplossing: GPU
  - CPU zet opdracht in VRAM
  - GPU voert dit uit
  - GPU kan: lijnen, cirkels, polygonen, 3D, ...

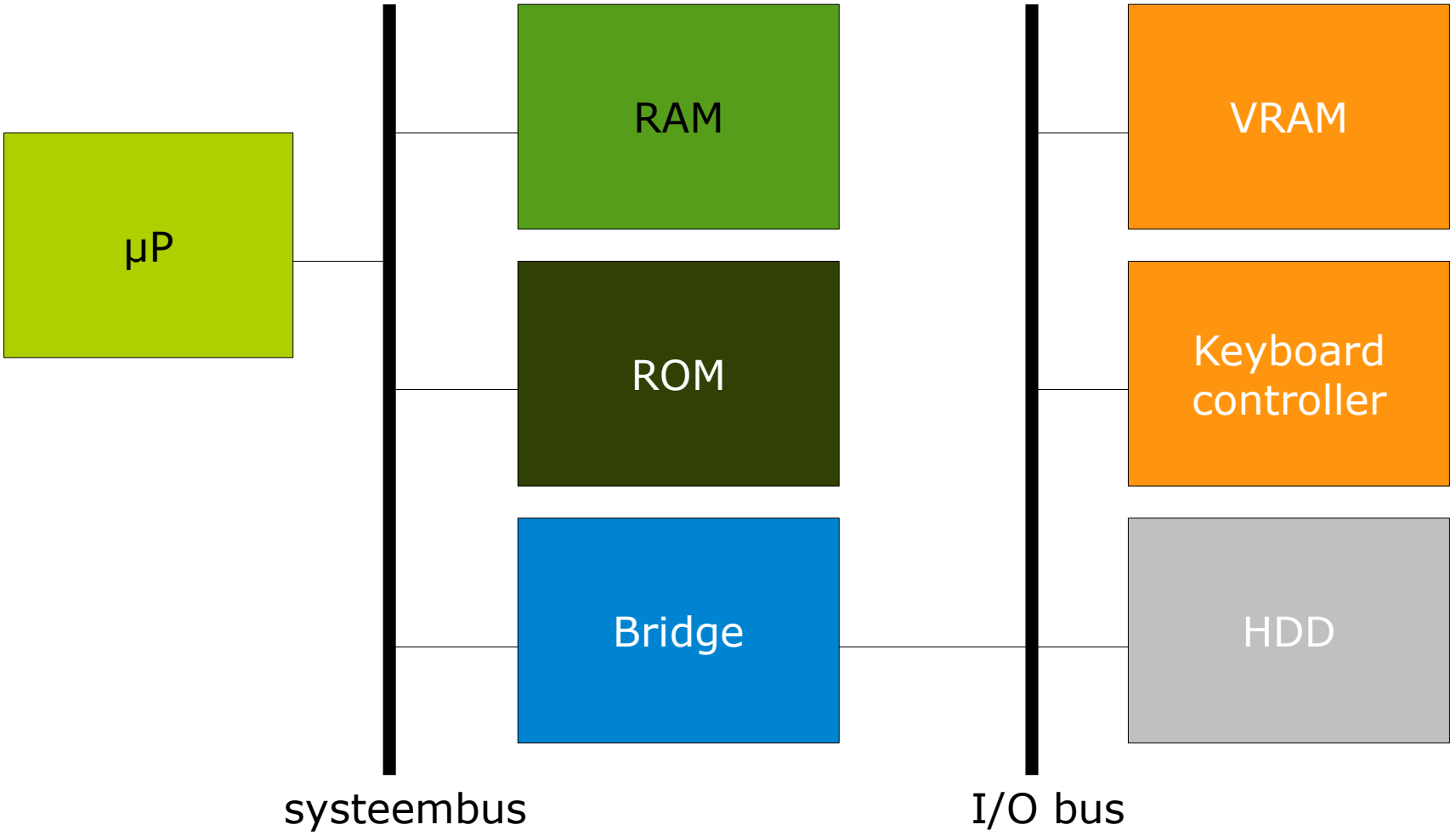
Inhoud

- Beeldscherm
- Toetsenbord
- Disk drive
- Herhalingsvragen

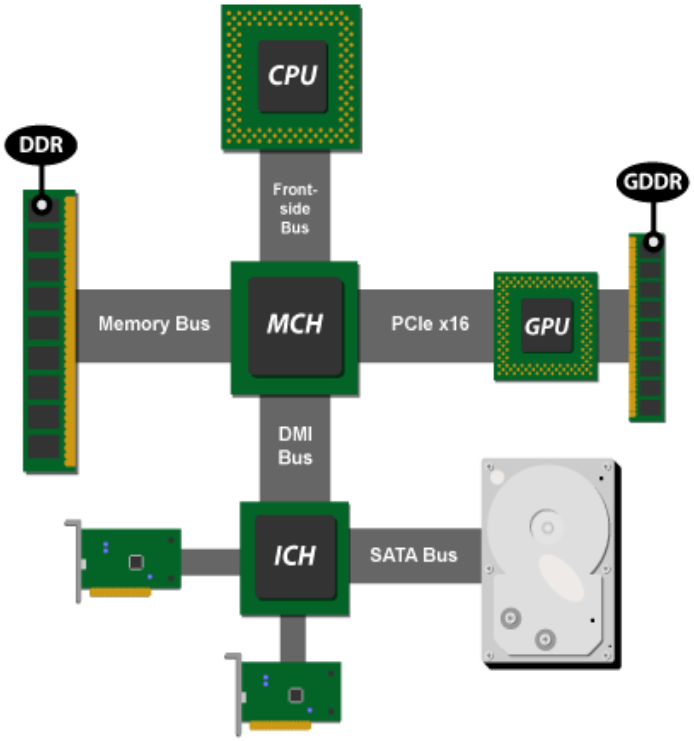
Input/Output - Toetsenbord

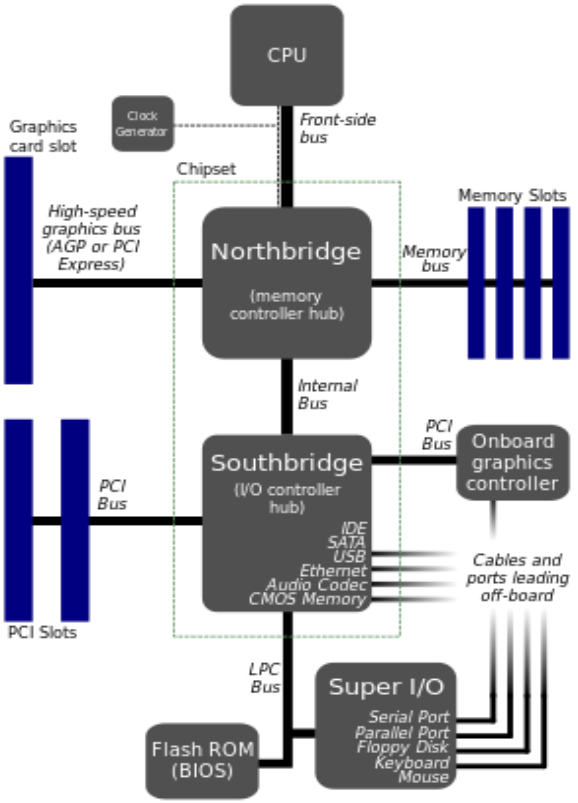


Toetsenbord – Memory-mapped



Bridges en bussen

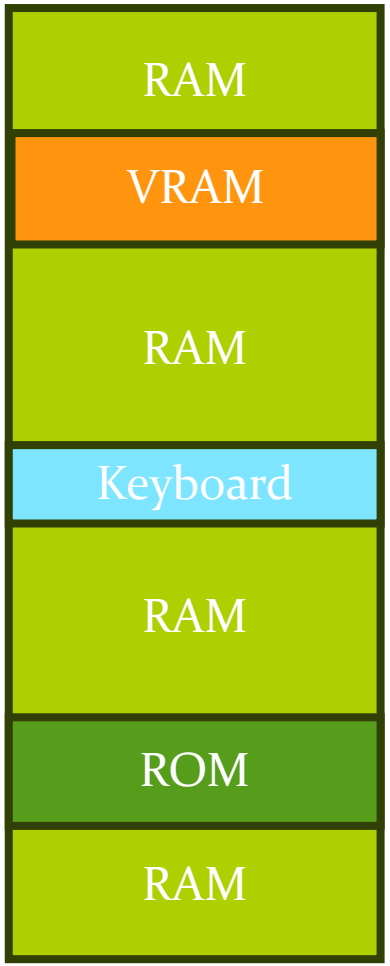




Toetsenbord – Memory-mapped

register keyboard != register CPU

- keyboard controller bevat 3 "registers"
  - data (laatste aanslag)
  - status (is er data klaar?)
  - control (settings)
- keyboard "registers" worden in geheugen gemapt: memory-mapped I/O*



*** Het grote voordeel van memory mapped I/O is dat de gewone ld en st machinetaal instructies gebruikt kunnen worden om I/O te doen**

Programmed I/O

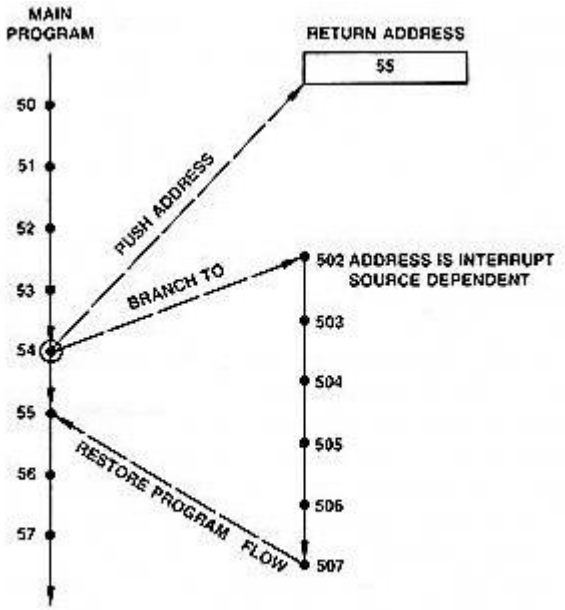
- stel: programma wacht op toets

```
int status;
do {
  status = memory[KEYB_STATUS];
} while (status != GEREED)
char c = memory[KEYB_DATA];
```

- polling
- dit heet programmed I/O
- nadeel?

Interrupt driven I/O

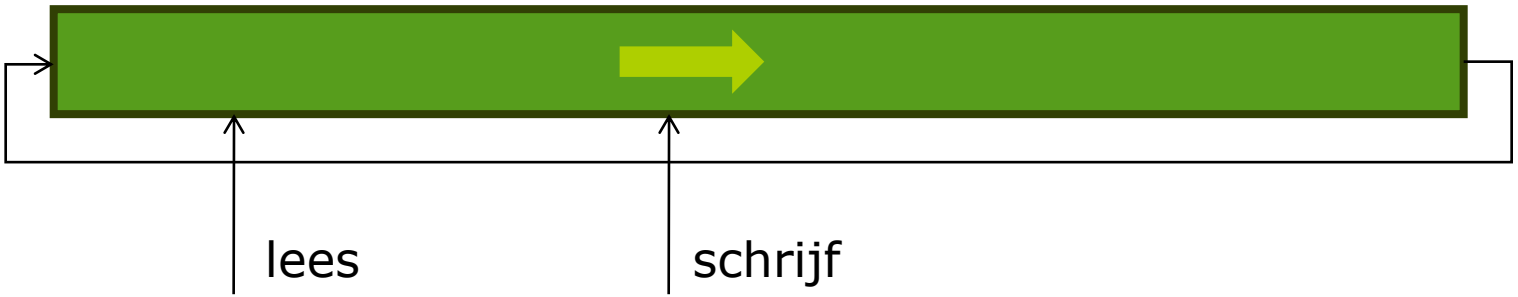
- het zou beter zijn als de processor een sein krijgt als er data klaar staat -> interrupt
  - Pin op de CPU
  - CPU krijgt interrupt van keyboard controller wanneer op toets wordt gedrukt
  - Interrupt tabel: interrupt nr. <-> adres ISR
  - Interrupt Service Routine (ISR) wordt uitgevoerd
  - Nadien teruggesprongen naar oorspronkelijke taak



Interrupt driven I/O

Circulaire buffer

- ISR zal toetsaanslagen in "circulaire buffer" in RAM zetten
- Circulaire buffer
  - Vaste grootte
  - lees=schrijf => empty
  - schrijf=lees-1 => full + risk for overflow

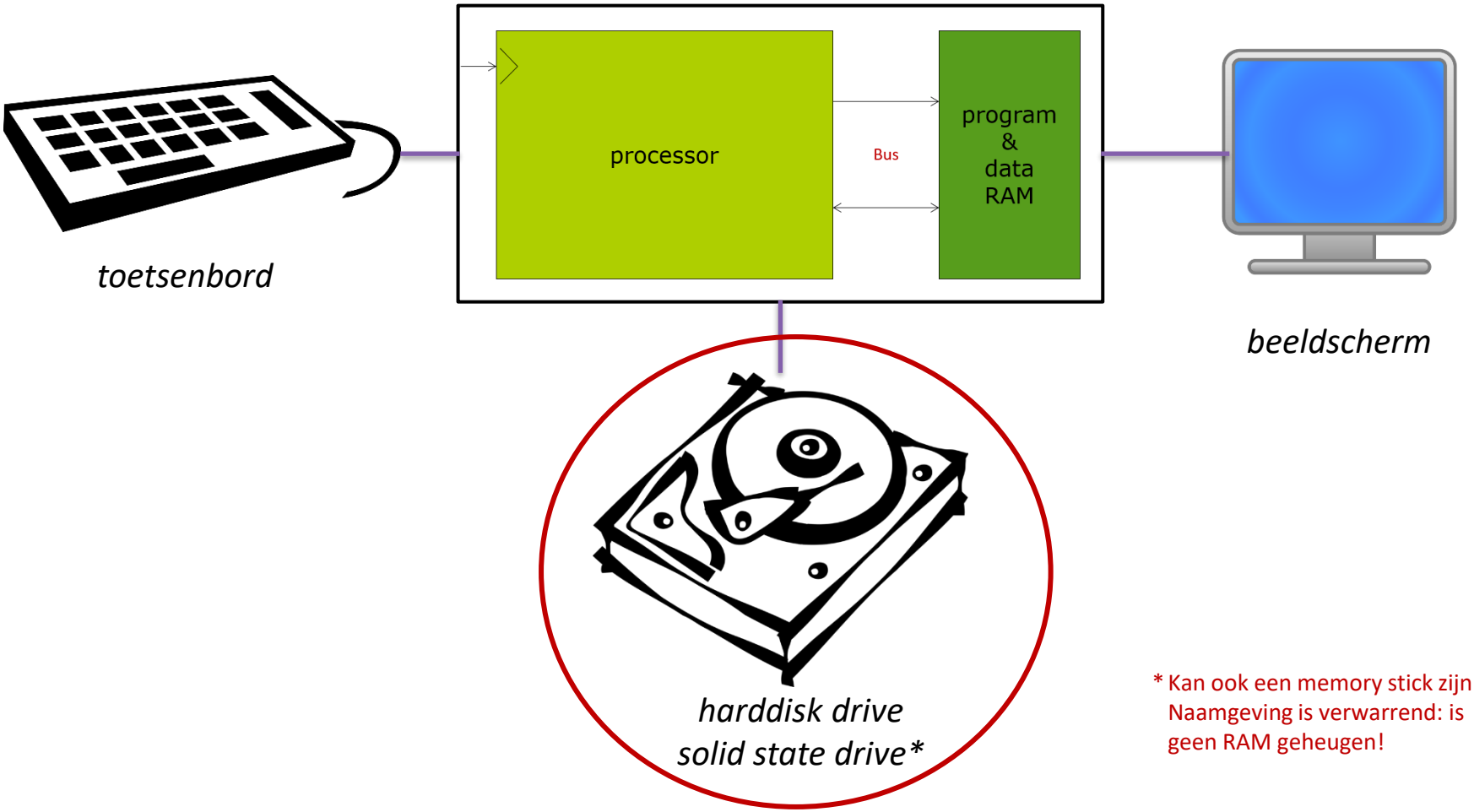


- software zal buffer af en toe lezen

Inhoud

- Beeldscherm
- Toetsenbord
- Disk drive
- Herhalingsvragen

Input/Output - Disk drive

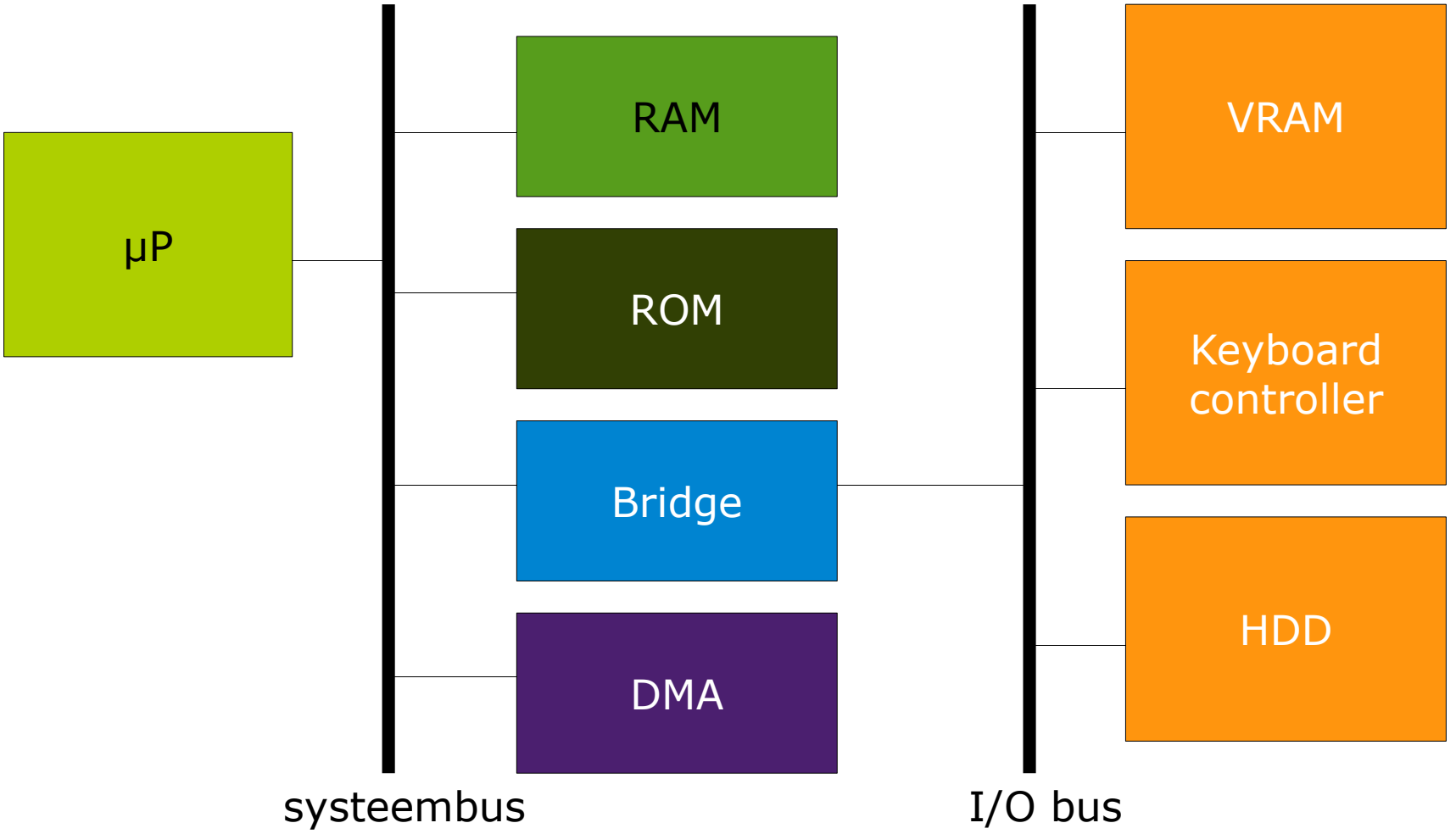


* Kan ook een memory stick zijn
Naamgeving is verwarrend: is geen RAM geheugen!

Disk drive

- Interrupt driver I/O veel efficiënter dan programmed I/O
- Bij lezen van disk worden grote hoeveelheden data verplaatst
- Wat is probleem?
- --> Direct Memory Access (DMA)
- Processor geeft aan DMA opdracht om data te lezen (of schrijven) en kan ondertussen iets anders doen
- DMA krijgt op dit moment controle over de bus!

DMA


µP
RAM
ROM
Bridge
DMA
systeembus
VRAM
Keyboard controller
HDD
I/O bus

Inhoud

- Beeldscherm
- Toetsenbord
- Disk drive
- Herhalingsvragen

herhalingsvragen

- Welke componenten zitten er op een video-kaart?
- Wat is de rol van de GPU?
- In welke modes kan een video kaart (grosso modo) werken?
- Stel dat een grafische kaart in ... mode staat met een resolutie van ... De vram is gemapt op ... Op welke plaats in het RAM geheugen moet wat geschreven worden om een pixel/karakter te tekenen?
- Wat is memory-mapped I/O? Wat zijn de voordelen ervan? Gebruiken alle computers memory-mapped I/O
- wat is het verschil tussen programmed I/O en interrupt-driven I/O? Wat is het voordeel van interrupt-driven I/O?
- Leg de werking van een circulaire buffer uit. Wanneer vol? Wanneer leeg?
- Wat is een ISR? wat doet het?
- Wat is een DMA controller? Wat doet deze? Waarom is die nodig?