Antwoorden week 3 – Microprocessorarchitectuur

In dit document vind je de antwoorden van de theorievragen voor week 3 van EMS20. Deze vragen kun je vinden op: https://bitbucket.org/HR_ELEKTRO/ems20/wiki/Theorie/Theorie_Week_3.pdf.

Het is verraderlijk om de onderstaande antwoorden in te zien, zonder eerst zelf de vraag te beantwoorden. Je denkt dan al snel: "inderdaad, dat is het juiste antwoord". Maar als je straks op de toets zelf het antwoord moet bedenken, dan blijkt een en ander toch niet zo eenvoudig te zijn.

Gebruik dit document alleen om je eigen antwoorden te controleren.

3.1

- Moderne microcontrollers hebben veel adreslijnen (bijvoorbeeld 32) en dus een grote geheugenruimte (memory map). In deze geheugenruimte is plaats genoeg voor de I/O registers.
- Bij een microcontroller bevinden de I/O poorten zich in dezelfde chip als de CPU. De selectielogica bevindt zich dus ook in dezelfde chip. Een paar extra logische poorten voor de selectielogica maakt dan niet meer uit.
- Als er geen speciale I/O-ruimte (I/O map) is, dan zijn er geen speciale instructies nodig voor I/O operaties.

3.2 **A** I/O mapped. Er wordt namelijk een VPA (Valid Peripheral Address) gebruikt.

B Direct: A_4 en A_5. Geïnverteerd: A_0, A_1, A_2, A_3, A_6 en A_7.

3.3 Om te voorkomen dat de databus veranderd als deze gelezen wordt door de microcontroller[1].

3.4 0x31.

3.5 **7.12** 2^4.

[1] We weten dan niet welke data wordt ingelezen. Maar er is nog een groter probleem. Het inklokken van veranderende data kan metastabiliteit veroorzaken, zie https://en.wikipedia.org/wiki/Metastability_(electronics).

3.6 De waarde van de PC wordt voor aanvang van de ISR opgeslagen op de stack en bij het verlaten van de ISR (bij het uitvoeren van de RTI-instructie) weer in de PC geladen.

3.7 De waarde van elk intern register wordt voor of bij aanvang van de ISR opgeslagen op de stack en voor of bij het verlaten van de stack wordt elk register weer geladen met de opgeslagen waarde.

3.8 Elke exception of groep van exceptions heeft een vast gereserveerde geheugenplaats waar het adres van de ISR die moet worden aangeroepen is opgeslagen. Deze geheugenplaats wordt meestal de exception vector genoemd.

3.9 De RET instructie haalt allen de PC van de stack. De RTI haalt alle registers van de stack die voor aanvang van de ISR op de stack zijn geplaatst. Dat zijn minstens het statusregister en de PC.

3.10 Als een ISR onderbroken kan worden door een andere interrupt dan moeten alle werkregisters, het statusregister en de PC weer op de stack worden geplaatst. Als de volgende ISR dan weer onderbroken kan worden door de volgende interrupt dan is er veel stackruimte nodig. Dit kan bij microcontrollers met weinig RAM (zoals de MSP430) leiden tot een stack overflow.

Als een ISR onderbroken kan worden door een andere interrupt, dan zorgt dit ervoor dat de interrupt die als eerste is opgetreden uiteindelijk als laatste (helemaal) wordt afgehandeld. Dat is in de meeste gevallen niet wenselijk.

3.11 I/O-3. Deze ontvangt als eerste de INT-ACK.

3.12 Dan wordt de ISR onderbroken en wordt eerst de NMI afgehandeld (dus wordt eerst de NMI-ISR uitgevoerd).

3.13 Von Neumann. De CPU heeft slechts één adres- en databus.

3.14 **A** Harvard-architectuur. De processor heeft een aparte code- en data-interface.

B De interrupt controller maakt gebruik van interrupt vectoren waarbij elke interrupt zijn eigen vast gereserveerde geheugenplaats heeft (zijn interrupt vector) waar het adres van de betreffende ISR moet worden opgeslagen.

C Bepaalde interrupts kunnen andere interrupts onderbreken.

3.15 Microcontroller.

3.16 **A** 256 plaatsen.

B 2 plaatsen.

C minder plaatsen nodig in de memory map.

D twee memory cycles nodig in plaats van één om één register te lezen of te schrijven.

3.17 ADC. Die heeft DMA.

3.18 Er is een busarbiter nodig omdat er twee potentiële busmasters zijn (de CPU en de DMAC).