

Theorie week 3 – Microprocessorarchitectuur

Deze week werk je verder aan leerdoel 1 dat is beschreven in de cursushandleiding.

Je gaat deze les leren:

- hoe de CPU met de buitenwereld communiceert via I/O-poorten en hoe deze I/O-poorten zijn opgebouwd;
- hoe de normale uitvoering van een programma onderbroken kan worden door verschillende exceptions, waaronder interrupts;
- hoe een interrupt controller is opgebouwd en werkt;
- hoe DMA werkt en waarom je dit zou willen toepassen in een embedded systeem.

Studiemateriaal week 3

De theorie die je deze week moet bestuderen kun je vinden in de volgende paragrafen van je boek¹:

- 7.5 I/O
- 7.6 Exceptions
- 7.7 Harvard-architectuur
- 8.3 I/O-poorten (en DMA)

Verdieping week 3

De in deze paragraaf besproken onderwerpen bieden verdieping aan op de in het boek gegeven informatie. De in deze paragraaf besproken onderwerpen behoren echter niet tot de toetsstof.

Op pagina 147 van je boek wordt een probleem vermeldt van de schakeling in figuur 7.46: “Een tweede probleem kan zijn dat de data net veranderen als de CPU deze gaat lezen”. Waarom is dat een probleem? Tip: bedenk dat de data afkomstig van de inputpoort wordt ingeklokt in een general purpose register en zie [https://nl.wikipedia.org/wiki/Metastabiel_\(elektronica\)](https://nl.wikipedia.org/wiki/Metastabiel_(elektronica)).

Op pagina 149 van je boek staat: “De CPU kan op een interrupt op de volgende wijze reageren: Zet alle werkregisters, de program counter en het statusregister (van de CPU) op de stack.

¹ Leo van Moergestel. *Computersystemen en embedded systemen*. 4de ed. Boom, 2016. ISBN: 9-789058-754233.

Enzovoorts”. De meeste processoren plaatsen de werkregisters niet automatisch op de stack. Bijvoorbeeld de MSP430, die je bij EMS10 hebt gebruikt, zet bij een interrupt alleen de program counter en het statusregister op de stack. De overige registers moeten indien nodig door de assemblertaalprogrammeur of door de C-compiler in de ISR op de stack worden geplaatst. Bedenk wat het voordeel is van deze laatste methode.

Op pagina 150 wordt uitgelegd hoe een software interrupt werkt. Er wordt echter niet verteld waarvoor een software interrupt over het algemeen wordt gebruikt. Zoek uit waarvoor software interrupts voor worden gebruikt. Tip: zie paragraaf 12.4 van je boek.

Opgaven week 3

Gebruik bij het maken van de onderstaande opgaven in eerste instantie alleen je boek. Maak alleen als je er met behulp van het boek niet uitkomt, gebruik van andere bronnen zoals het internet. Bij de toets mag je immers ook alleen je boek maar gebruiken.

3.1 Geef minstens twee goede redenen waarom moderne microcontrollers bijna altijd memory mapped I/O gebruiken terwijl verouderde processoren zoals de Z80 I/O mapped I/O gebruikten.

3.2 Zie figuur 7.45 op pagina 146 van je boek.

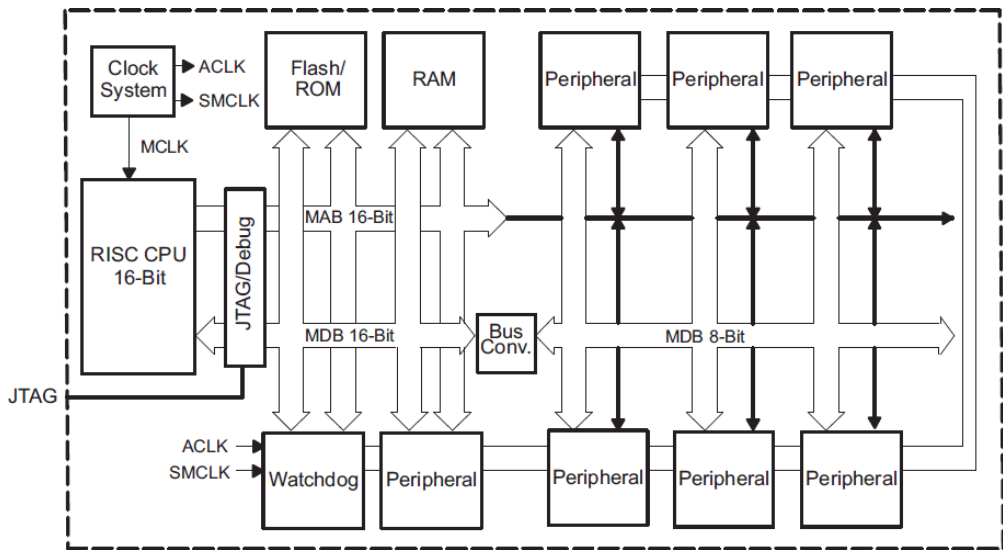
A Is dit een memory mapped of een I/O mapped outputpoort? Verklaar je antwoord.

B We gaan er bij deze deelvraag van uit dat deze poort is aangesloten op een CPU die 8 adreslijnen gebruikt om een I/O-poort te selecteren (zoals bijvoorbeeld de Z80). We willen de output poort kunnen beschrijven op adres 0x30. Welke adreslijnen moeten dan direct en welke adreslijnen moeten dan geïnverteerd op de AND-poort worden aangesloten?

3.3 Zie figuur 7.47 op pagina 147 van je boek. Waarom is de AND-poort voor de klokingang van het buffer nodig?

3.4 Zie figuur 7.48 op pagina 148 van je boek. Als het inputdataregister zich bevindt op adres 0x30, wat is dan het adres van het bijbehorende statusregister?

- 3.5** Maak 7.12 op pagina 157 van je boek. Deze vraag heeft betrekking op figuur 7.58 dat is weergegeven op diezelfde pagina.
- 3.6** Hoe kan het dat de CPU na het afhandelen van een ISR automatisch verdergaat waar het uitvoeren van instructies werd onderbroken?
- 3.7** Hoe kan het dat het onderbroken programma er niets van merkt dat tussendoor een ISR is uitgevoerd?
- 3.8** Hoe weet de CPU welke ISR uitgevoerd moet worden als er een bepaalde exception optreedt?
- 3.9** Wat is het verschil in gedrag tussen de return from interrupt (RTI) instructie (die gebruikt wordt in figuur 7.49) en de return from subroutine (RET) instructies (die gebruikt wordt in figuur 7.39)?
- 3.10** Op pagina 153 van je boek staat: “Een CPU maakt zich bij het afhandelen van een interrupt meestal tijdelijk ‘doof’ voor het interrupt-sigitaal”. Bij EMS10 heb je gezien dat dit bij de MSP430 ook zo werkt. In het boek wordt echter niet uitgelegd waarom dit zo werkt. Bedenk zelf waarom het niet handig is als een interrupt onderbroken kan worden door een andere interrupt.
- 3.11** Beschouw figuur 7.55 op pagina 154 van je boek. Welk device van de 4 getekende devices heeft de hoogste prioriteit? Verklaar je antwoord.
- 3.12** Wat gebeurt er als tijdens het uitvoeren van een ISR een non-maskable interrupt optreedt?
- 3.13** In [figuur 1](#) is de architectuur van de MSP430 microcontroller weergegeven, die je bij EMS10 hebt gebruikt. Is dit een Von Neumann- of een Harvard-architectuur? Verklaar je antwoord.



Figuur 1: De architectuur van de MSP430 microcontroller. Bron: [MSP430x2xx Family User's Guide](#).

3.14 In [figuur 2](#) is de architectuur van de Cortex-M4 processor weergegeven. Deze processor is een onderdeel van de CC3220S SoC die je in het practicum gebruikt.

A Is dit een Von Neumann- of een Harvard-architectuur? Verklaar je antwoord.

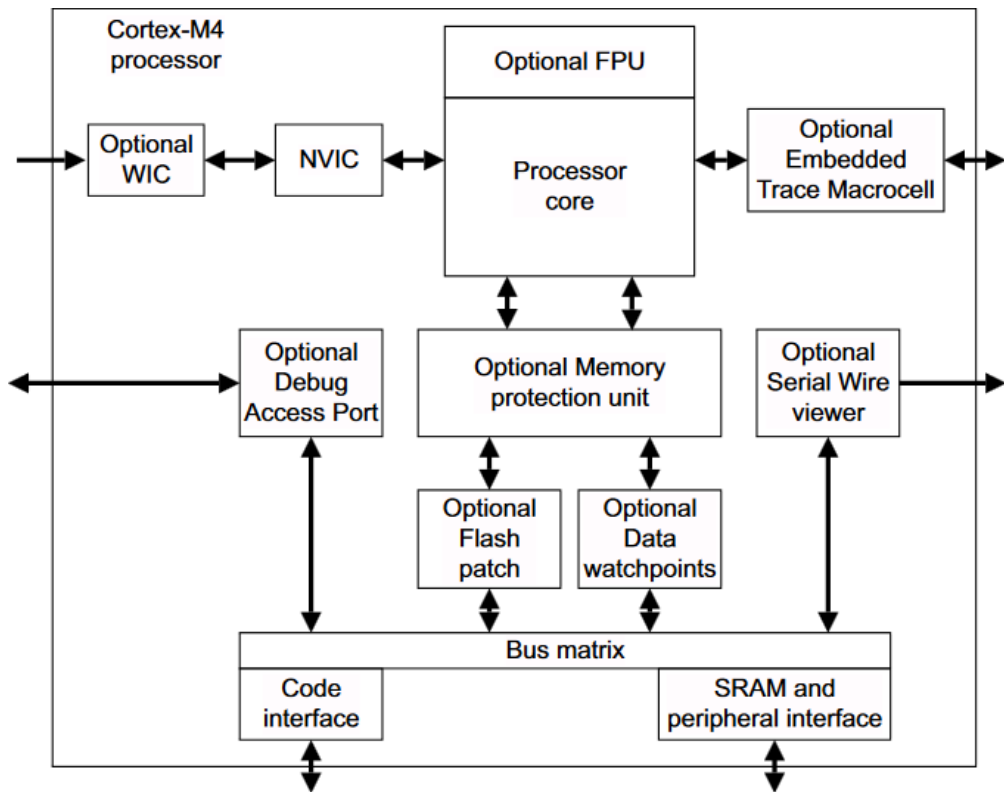
De afkorting NVIC staat voor Nested Vectored Interrupt Controller. Deze naam zegt iets over de eigenschappen van de interrupt controller van de Cortex-M4.

B Welke eigenschap van de interrupt controller leid je af uit de term “Vectored”?

C Welke eigenschap van de interrupt controller leid je af uit de term “Nested”?

3.15 Op pagina 163 van je boek staat: “I/O kan op dezelfde chip geïntegreerd zijn als de CPU maar ook als afzonderlijke component beschikbaar zijn”. Hoe noemen we een chip waarop de CPU, I/O en memory geïntegreerd zijn?

3.16 Op pagina 164 wordt besproken dat een I/O-module geconfigureerd moet worden door de CPU. Er worden daarbij een aantal mogelijkheden opgesomd over hoe de CPU



Figuur 2: De architectuur van de Cortex-M4 processor. Bron: *Cortex™-M4 Generic User Guide*.

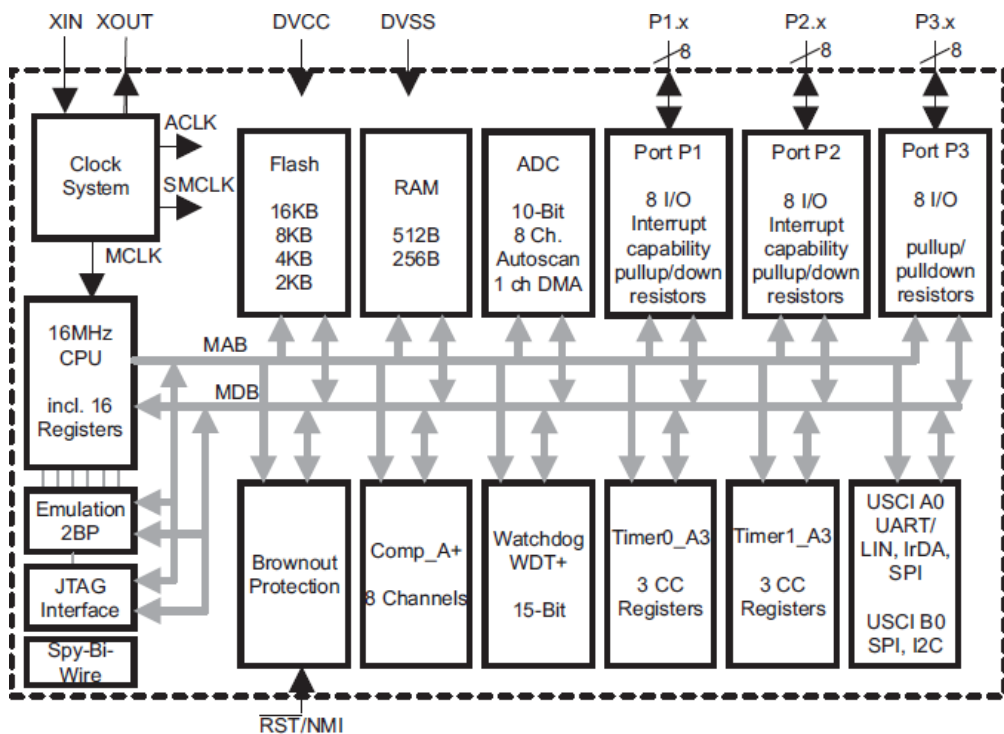
met de I/O-module kan communiceren via controleregisters². Stel dat een complexe I/O-module intern 256 controleregisters bevat.

- A** Hoeveel ruimte neemt deze I/O-module in, in de memory map, als elk controle-register afzonderlijk door de CPU geadresseerd kan worden?
- B** Hoeveel ruimte neemt deze I/O-module in, in de memory map, als gebruik wordt gemaakt van een zogeheten pointer-register?
- C** Wat is het voordeel van het gebruik van een pointer-register ten opzichte van apart adresseerbare controleregisters?

² In navolging van het boek wordt hier het woord “controleregister” gebruikt. Dit is in feite een verkeerde vertaling van de Engelse term “control register”. Er wordt immers niets gecontroleerd (nagekeken). Een betere vertaling zou “besturingsregister” zijn.

D Wat is het nadeel van het gebruik van een pointer-register ten opzichte van apart adresseerbare controleregisters?

3.17 In [figuur 3](#) is de architectuur van de MSP430G2553 microcontroller weergegeven, die je bij EMS10 hebt gebruikt. Één van de peripherals (I/O-modules) van deze microcontroller is in staat om rechtstreeks (zonder tussenkomst van de CPU) data in het RAM te schrijven. Welke peripheral is dit? Verklaar je antwoord.



Figuur 3: De architectuur van de MSP430G2553 microcontroller. Bron: [MSP430G2x53](#), [MSP430G2x13 Mixed Signal Microcontroller datasheet](#).

3.18 Zie figuur 8.5 op pagina 165 van je boek. Heeft dit systeem een *busarbiter* nodig? Verklaar waarom wel/niet?