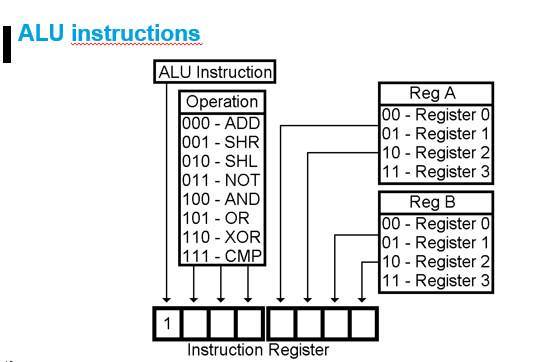
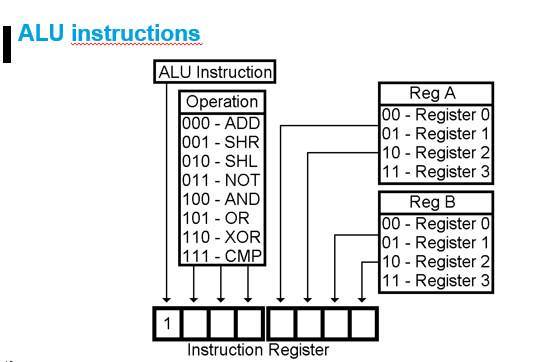
Zorg ervoor dat je, voordat je aan deze vragen begint, tot en met bladzijde 140 in het boek hebt gelezen.

1. Beschrijf de functie van de volgende onderdelen:
   1. Instruction Register
      1. The bits from this register will "instruct" the CPU what to do.
      2. Houdt de instructie tijdelijk vast voor de cpu.
   2. Instruction Address Register
      1. to store the RAM address of the next instruction that we want to move into the IR. If the IAR contains 0000 1010 (10 decimal,) then the next instruction that will be moved to the IR is the byte residing at RAM address ten.
      2. Bevat de locatie van de huidige instructie. Ook wel bekend als Program counter.
   3. Clock
      1. We need to turn the appropriate control bits on and off at the appropriate times.
      2. Ritme of snelheid van de computer aan. Dit doet hij door een puls.
   4. Stepper
      1. A stepper has two inputs: a clock and a reset. For output is has a number of bits, each of which will come on for one clock cycle.
      2. Helpen de volgorde met stappen, deze code heeft eigen output. Je moet weten dat er zes of zeven stappen zijn. Je hebt meerkeuze je moet wel weten hoe bijvoorbeeld de stappen in elkaar zitten.
   5. RAM
      1. Het tijdeljik opslaan van
      2. Het werkgeheugen, kortetermijn geheugen.
   6. MAR
      1. Hier staat de locatie van het benodigde ding in het ram.
      2. Waar de benodigde data te vinden, van het ram.
2. Wat is "fetching" met betrekking tot een instructie, en wat is "executing"?
   1. The stepper's first three steps are shown here, and result in 'fetching' the next 'instruction' from RAM. Then the rest of the steps 'execute' the 'instruction.' Exactly what will be done in steps 4, 5 and 6, is determined by the contents of the instruction that was fetched. Then the stepper starts over, fetches the next instruction, and executes it.
   2. Fetching: zoekt de nieuwe instructive op.
   3. Execution: voert hij hem uit
3. De volgende byte-waarden zijn instructies voor de CPU uit het boek. Wat wordt er uitgevoerd door de ALU? Geef aan welke registers betrokken zijn, welke actie wordt uitgevoerd en in welke register het antwoord wordt geplaatst. De eerste vraag is voorgedaan. 
   1. 1001 1011  
      **Antwoord:** De byte-waarde uit register 2 wordt geshift naar links. Het antwoord wordt in register 3 geplaatst. (is het niet naar rechts?)
   2. 1100 0001
      1. De waardes uit areg0 en Breg1 worden geAND met elkaar, waarna het antw in Breg1 wordt geplaats.
   3. 1011 1010
      1. De byte-waarde uit Aregister 2 wordt genot en het antwoord wordt geplaats in Bregister2 (kan dit uberhaupt omdat het reg 2heet)
   4. 1000 1110
      1. De byte-waardes uit Areg3 en Breg2 worden bij elkaar opgeteld (ADD) en het antwoord wordt verzonden naar Breg 2
   5. 1000 0000
      1. De waardes uit Areg0 en Breg0 worden bij elkaar geteld (ADD) en geplaatst in Breg0
4. Schrijf de instructies van opdracht 3 in de computertaal (Assembly language) van pagina 122 uit het boek.
   1. 1001 1011
      1. SHR, R2 R3
   2. 1100 0001
      1. AND R0 R1
   3. 1011 1010
      1. NOT R2 R2
   4. 1000 1110
      1. ADD R3 R2
   5. 1000 0000
      1. ADD R0 R0
5. De instructies van opdracht 3 worden per ongeluk in het verkeerde gedeelte van het geheugen gelezen. De byte-waarden zijn geen instructies, maar gewoon getallen. Geef van elke byte-waarde de decimale waarde en de hexadecimale waarde.
6. 1001 1011
   1. 155
   2. 0x9B
7. 1100 0001
   1. 193
   2. 0xC1
8. 1011 1010
   1. 186
   2. 0xBA
9. 1000 1110
   1. 142
   2. 0x8E
10. 1000 0000
    1. 128
    2. 0x80
11. Het vertalen van de byte-waarden uit opdracht 3 naar ASCII levert geen zinnige waarden op. Waarom niet? Gebruik het internet om deze vraag te beantwoorden.
    * 1. Omdat het gaat om een ALU instructie en niet een ascii code.
      2. Ascii gat van 0 tot 127 binair, en je ziet dat dit grotere getallen zijn en daardoor passen ze er niet in.
12. Geef de instructiecodes (bit-waarden) van de volgende instructies in de computertaal (Assembly language) uit het boek. Geef ook aan wat de CPU doet als deze de instructiecode uitvoert. (oke is dit een toets vraag of niet?)
    1. 100



* 1. SHR R1, R2
     1. 1001 0110
  2. ST R0, R3
     1. 001 0011
  3. JMP; 203
     1. 0100000 11001011
  4. CLF
     1. 0110 0000
  5. CMP R2, R3
     1. 1111 1011
  6. DATA R2; 14
     1. 00100010 00000111
     2. 00100010 00001110
  7. JMPR R1
     1. 00110001
  8. JE; 156
     1. 01010010 10011100
  9. XOR R1, R3
     1. 1110 0111
  10. JCZ; 89
      1. 0101 1001 1011001
      2. 01011001 01011001

1. Beschrijf per stap van de stepper wat de CPU doet als in het instructieregister de bytewaarde 11001101 zit.
   * 1. Als eerste bit 1 is dan is het een alu.

1. De locatie van de instructie wordt in de MAR geladen, het adresregister wordt met 1 verhoogt.

2. De instructie komt uit het ram en gaat het instructieregister in

3. Het antwoord van het verhogen van het adresregister met 1 (zie stap 1) komt uit de ACC(accumulator) en gaat het IAR (instruction address register) in. (CPU gaat nu uiteraard verder met het uitvoeren van de instructie in de IR, maar is in principe terug bij stap 1, er staat een nieuw adres voor een instructie klaar)

4. De inhoud van register B wordt op de bus gezet en naar TMP gestuurd

5. De inhoud van register A wordt op de bus gezet, de ALU krijgt zijn instructies en de output gaat naar ACC (ALU heeft op dit moment toegang tot TMP en register A (die op de bus staat)).

6. De inhoud van ACC wordt op de bus gezet en naar register B gestuurd.

7. De stepper reset.

*Je hoeft de antwoorden niet in te leveren, deze worden in een les behandeld.*