

Antwoorden tentamenvragen hoofdstuk 7

- Leg uit op welke manier de executie van een 'loopje' precies plaatsvindt. Beschouw hierbij de machine afgebeeld in figuur 7.5.

Antwoord: Een sprong vindt plaats als de uitgang van de AND-poort 1 is. Hiervoor gelden twee voorwaarden: Er is een branch-instructie (Branch = 1) en de waarde op de ALU-uitgang is 0 (Zero = 1). De uitgang van de AND-poort is verbonden met LoadPC. Op de eerstvolgende opgaande klokflank wordt de PC geladen met de waarde op de uitgang van de Adder. Deze waarde is de som van de huidige waarde van de PC + de offset die in het instructiegeheugen is opgeslagen.

- Geef een expressie in termen van Boole-algebra voor de ALU-uitgang Zero.

$$Z = y_0 + y_1 + y_2 + \dots + y_{15}$$

- Beschouw figuur 7.5.

- Waar is de component "Adder" voor nodig?

Antwoord: Om het nieuwe adres van de PC te berekenen.

- Op welk adres staat de volgende instructie die wordt uitgevoerd?

Antwoord: Op adres 0x005.

- Welke operatie voert de ALU uit?

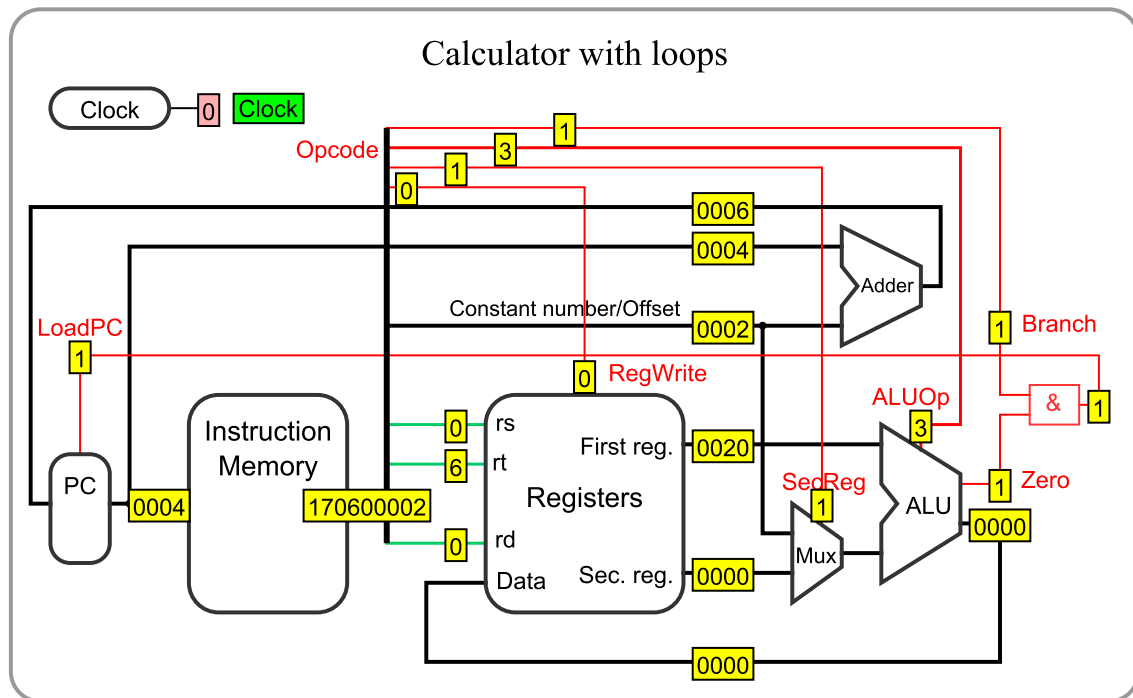
Antwoord: De waarde op ingang B doorlaten.

- Welke instructie wordt uitgevoerd? Geef de syntax ervan weer.

Antwoord: BZ \$6, 6.

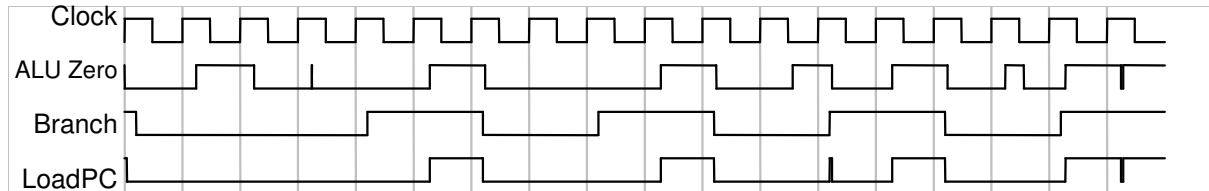
- Waarvoor is een "RegWrite"-ingang nodig bij de component registers?

Antwoord: Branch-instructies mogen niet naar één van de registers schrijven, dit in tegenstelling tot de andere instructies. Om dat mogelijk te maken is een "RegWrite"-ingang nodig.



Figuur 7.5

4. Beschouw het tijdvolgordediagram van figuur 7.6.
 - a. Hoeveel keer wordt een branch-instructie aangeroepen?
Antwoord: Bij klokpuls 5, 6, 9, 10, 13, 14, 17 en 18 dus 8 keer (klokpuls 18 is een HALT-instructie).
 - b. Hoeveel keer springt de PC hierdoor naar een andere instructie dan de eerstvolgende?
Antwoord: Bij klokpuls 6, 10, 14, 17 en 18 dus 5 keer



Figuur 7.6 Time Sequence diagram van een programma

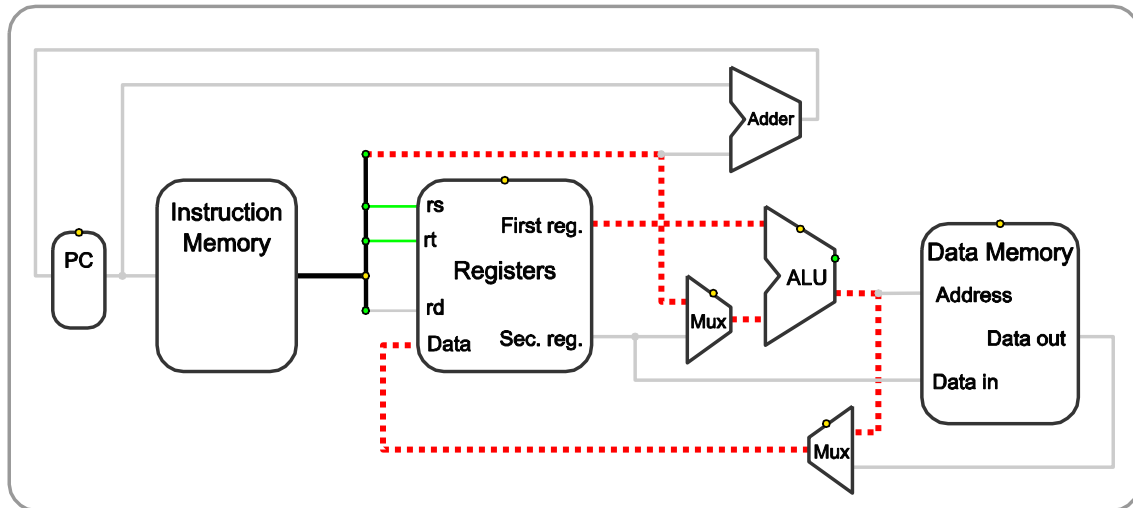
5. Waarom kan een instructie BEQI (Branch if Equal Immediate), bijvoorbeeld BEQ \$4, 5, label op de machine van figuur 7.5 niet worden geïmplementeerd?
Antwoord: Bij de instructie BEQI worden twee registerwaarden vergeleken en als deze gelijk zijn wordt er naar een 'label' gesprongen. De syntax zou moeten zijn: BEQI rs, rt, offset, label. De mux SecReg kan niet zowel rt als de offset doorlaten.

Antwoorden tentamenvragen hoofdstuk 8

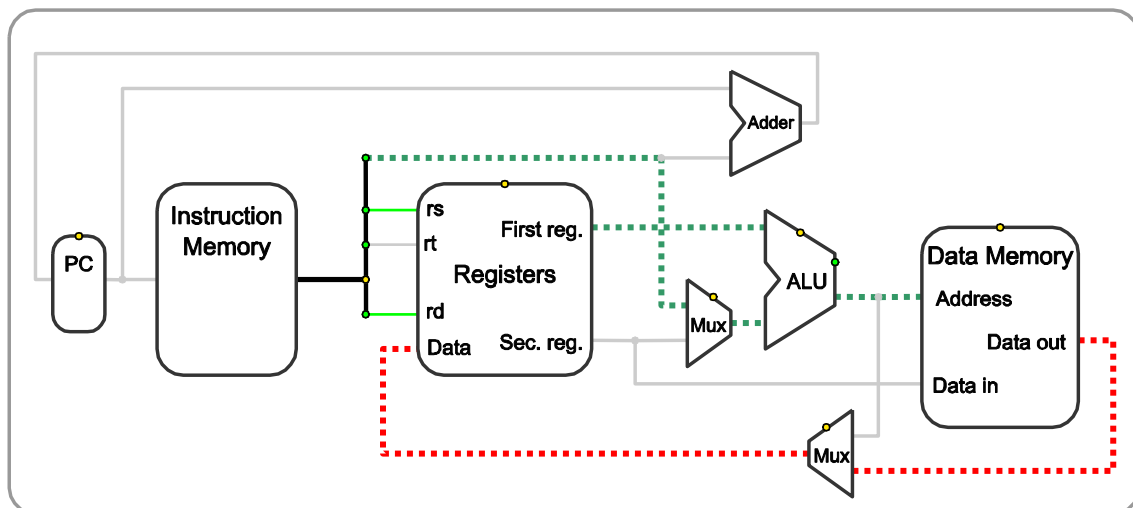
1. Uit welke vijf hoofdcomponenten is een Harvard machine opgebouwd? Geef de functie van elke component weer.
Antwoord:
 - 1: Program Counter. Deze bepaalt het instructieadres.
 - 2: Instruction Memory. Hier worden de instructies in opgeslagen.
 - 3: Registers. Hier worden (tussen)resultaten in bewaard.
 - 4: ALU. Het rekenorgaan van de processor.
 - 5: Data Memory. Hier worden de gegevens bewaard.
2. Waarom wordt de machine uit dit hoofdstuk een Load/Store machine genoemd?
Antwoord: De communicatie met het Data Memory vindt plaats met slechts twee instructies: Load Word en Store Word.
3. Wat is het 'Stored Program Concept'?
Antwoord: Bij het 'Stored Program Concept' worden instructies voorgesteld als getallen en programma's kunnen worden opgeslagen in het geheugen en worden geschreven en gelezen als getallen.
4. Uit welke fases bestaat het executeren van een "Store Word"-instructie.
Antwoord: Instruction fetch, instruction decode, execution en data memory access.
5. Wat is de opcode van een instructie?
Antwoord: Het veld van de instructie dat bepaalt welke instructie wordt uitgevoerd.
6. Geef het basisprincipe weer waarop een pipeline machine is gebaseerd.
Antwoord: Bij een pipeline machine worden meerdere instructies tegelijkertijd uitgevoerd. Hierbij wordt gebruik gemaakt van het feit dat één instructie wordt uitgevoerd in meerdere fases. Eerst wordt de eerste fase van de eerste instructie uitgevoerd. Hierna worden tegelijkertijd de tweede fase van de eerste instructie en de eerste fase van de tweede instructie uitgevoerd. Daarna worden tegelijkertijd de derde fase van de eerste instructie, de

tweede fase van de tweede instructie en de eerste fase van de derde instructie tegelijkertijd uitgevoerd en zo verder.

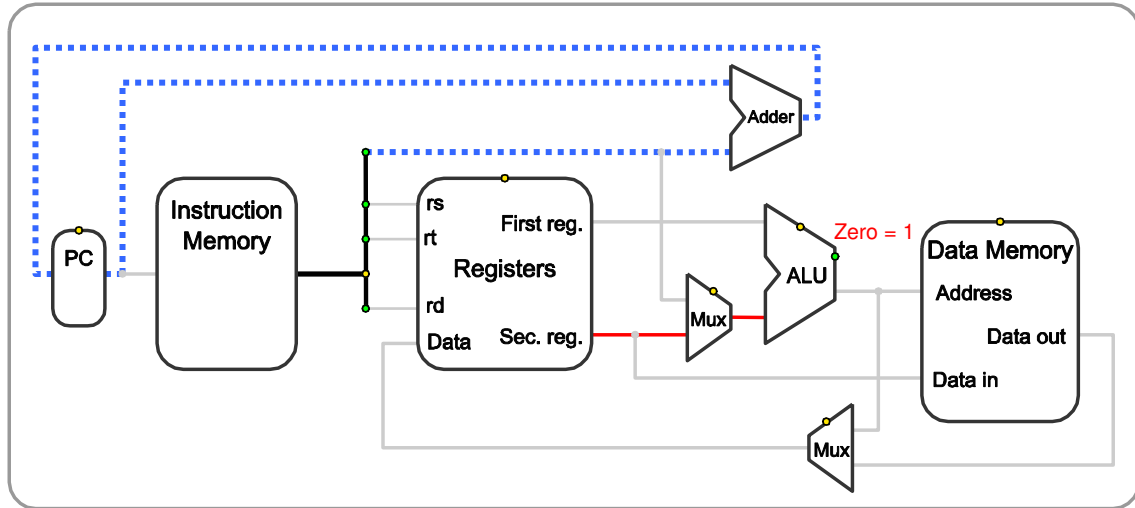
7. In figuur 8.5 is een machine weergegeven met alleen de data- en adreslijnen tussen de verschillende componenten dus zonder de controlelijnen.
- Arceer het datapad en het adrespad van een ADDI-instructie.
 - Arceer het datapad en het adrespad van een Load Word-instructie.
 - Arceer het datapad en het adrespad van een Branch Zero-instructie.



ADDI-instructie

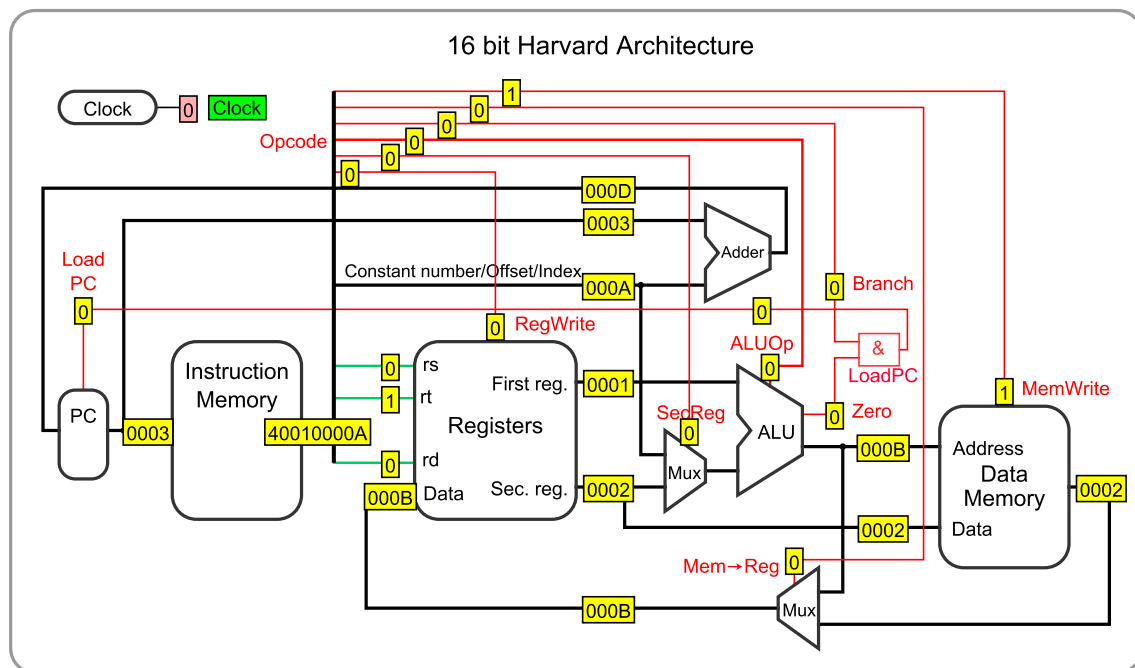


Load Word-instructie



Branch Zero-instructie

8. In figuur 8.6 is een Harvard processor weergegeven. In deze figuur staat de term: Constant number/Offset/Index.
- Wat wordt bedoeld met "Constant number"? Bij welke instructies is deze term van toepassing?
Antwoord: Constant getal. Bij instructies van het type immediate.
 - Wat wordt bedoeld met "Offset"? Bij welke instructies is deze term van toepassing?
Antwoord: Offset van de PC t.o.v. de huidige stand. Bij branch-instructies.
 - Wat wordt bedoeld met "Index"? Bij welke instructies is deze term van toepassing?
Antwoord: Index (van een lijst of array). Bij Load Word en Store Word-instructies.
 - Geef een voorbeeld van een instructie waarbij geen van de drie bovengenoemde termen van belang is.
Antwoord: Bij register-register transfers. Bijvoorbeeld de instructie ADD \$3, \$4, \$5.
 - Welke instructie wordt in figuur 8.6 uitgevoerd? Geef de syntax ervan weer.
Antwoord: De ALU-uitgang heeft de waarde 0x000B. Dit kan alleen de som zijn van 0x000A en 0x0001. MemWrite is 1 dus er wordt naar het Data Memory geschreven. De syntax is: SW \$1, 000A, \$0. De inhoud van register 1, het getal 2, wordt geschreven naar adres 0x000B.
 - Wat is de code in machinetaal van deze instructie?
Antwoord: 0x40010000A.
 - Wat is de volgende stand van de Program Counter?
Antwoord: 4 (Load PC = 0, dus er vindt geen sprong plaats).



Figuur 8.6