

1. De noodzaak van luisterbits.

Een en ander zal gedemonstreerd worden aan een bandlezer.

1.1. Communicatie tussen Jansen en de X8, vanuit de X8 bezien.

Voor een bepaalde bandlezer neem ik de volgende conventies aan (zie X8, nr 22) Jansen is zich bewust van 6 elementen in de X8, die daar speciaal aan deze bandlezer zijn toegekend:

Adres f0 voor de leegwijzer van het startmagazijn: $M[f0] = f1$ of $f1 + 2$

Adres f1, dwz. het beginadres van het startmagazijn; ik neem devolgende structuur van het startmagazijn aan: $M[M[f0]] =$ heersende startinformatie

$M[M[f0] + 1] =$ heersend klassewoord

Adres f2 voor startopdrachtenteller: $M[f2] = 0, 1$ of 2 .

Hiermee is geassocieerd de boolean Acflop F2: $F2 = (M[f2] \text{ positief})$

Adres f3 voor ingreenteller: $M[f3] = 0, 1$ of 2 .

Hiermee is geassocieerd de inflop F3: $F3 = (M[f3] \text{ positief})$

De qualificaties "acflop" en "Inflop" als boven gegeven zijn "X8-centrisch"; in "Jansen-centrische" terminologie zouden de twee qualificaties net andersom zijn.

Startinformatie bevat voor de lengte 9 bits en voor het adres 18 bits. Er is voorshands aangenomen, dat een speciale configuratie fungeert als de opdracht "herstel OK-toestand". Deze veronderstelling is niet essentieel.

1.2. Communicatie tussen Jansen en bandlezer, vanuit Jansen bezien.

Ik neem aan, dat er voor de sybchronisatie tussen Jansen en de bandlezer twee flip-flops in het spel zijn, IF en VB.

IF is "jansencentrisch" een ingreep-flipflop;

VB is "jansencentrisch" een actiefliplflop.

Jansen zal "VB := true" slechts uitvoeren op momenten, dat IF = false; Jansen signaleert daarmee, dat de bandlezer zijn volgende verrichting kan uitvoeren. Dit impliceert VB := false en na voltooiing de terugmelding IF := true.

De naam VB is een afkorting van "Vul B" omdat karaktertransport zal plaats vinden via een register, dat in een of ander pamphlet een keer B(and) genoemd is. Dit register is vulbaar door de bandlezer en uitleesbaar door Jansen.

Als er nu geen OKF en STF waren, dan was het allemaal niet zo ingewikkeld: dan doet Jansen in wezen de cyclus:

"P(IF); lees B en zend door; V(VB)"

Om nu OKF en STF in de beschouwing te betrekken (zoals ze in X8 nr 22 geïntroduceerd zijn) moet ik, dacht ik straks nog wat nadere veronderstellingen maken.

Ik citeer: "Het drukken op de groene knop heeft tot gevolg:

STF := RK or EB↓

Zolang EB = true wordt, desnoods bij herhaling:

if non STF then begin OKF := false; STF := true end
uitgevoerd.

Herstel-OK betekent OKF := true; if STF then wacht.

STF en OKF zijn in deze beschrijving de enige houdende elementen. Met de rode knop (STF := true) zijn hiermee alle manieren gegeven, waarop deze twee flip-flops beïnvloed kunnen worden. Ik meen voor Jansen een werkend schema te hebben, zonder dat STF tot Jansen hoeft door te dringen.

Ik veronderstel, dat Jansen wel op elk gewenst moment OKF kan uitlezen, maar niet zelf OKF kan zetten. Dit is nl. niet nodig en misschien zelfs ongewenst.

Er zijn twee verschillende verrichtingen, die Jansen van de bandlezer kan vragen en ook inderdaad aan de bandlezer moet overlaten, omdat ze willekeurig lang kunnen duren:

- a) een volgend karakter, dwz. wel iets nieuws in B en bandtransport
- b) herstel OK-toestand en "if STF then wacht", niet iets nieuws in B en geen bandtransport.

Ik neem aan, dat er zich in de bandlezer een boolean HOK (Herstel OK) bevindt, die via de plug door Jansen = true gezet kan worden als de OK toestand hersteld moet worden.

Als de bandlezer de passeeroperatie op VB uitvoert, raadpleegt hij HOK: normaal is HOK = false; dan wordt het volgende karakter in B afgeleverd en dit wordt door V(IF) afgerond; als daarentegen HOK = true, dan doet de bandlezer het herstel van de OK-toestand:

```
OKF:= true; L: if STF then MM goto L; HOK := false; V(IF)
```

HOK vertoont enige analogie met B inzoverre zetten en raadplegen in de tijd strict gescheiden zijn.

Opm 1. Als Jansen succesvol de passeeroperatie op IF gedaan heeft, moet Jansen kunnen detecteren of er nu een nieuw karakter in B is gekomen of niet, dwz. of de vorige verrichting van de bandlezer herstel-OK was of niet. Jansen kan dit ontdekken door analyse van de vorige (Meersende zo je wilt) startopdracht. Ordelijker, ~~Opzettelijker~~, sneller en waarschijnlijk iets duurder is het wanneer op de bandlezerplug uitleesbaar is, of er een nieuw karakter in B zit, maw. of IF:= true al dan niet uitgevoerd is op grond van een voltooide herstel-ok-opdracht.

Opm.2. Ook na voltooiing van de herstel OK-opdracht moet Jansen de OKF uitlezen: er kan immers opnieuw storing zijn ontstaan.

Opm.3. Er is aangenomen, dat Jansen 1 maal per karakter cyclus kijkt, of de OKF nog wel true is; vlak na deze constatering kan Jansen dan nog om de volgende heptade vragen; het is de bedoeling, dat hij die dan nog krijgt, maw. dat er in de bandlezer geen blokkade zit, die dat verhindert. Jansen neemt dan verder de verantwoordelijkheid op zich, om niet om meer karakters te vragen, voordat de OK-toestand hersteld is.

1.3. De programmering voor Jansen.

Ik neem aan, dat de individuele acties van Jansen zo kort duren, dat we hen in doofheid kunnen verrichten: als besloten is, dat een karakter van de bandlezer getransporteerd gaat worden, dan maakt Jansen die handeling eerts af, voordat hij gaat kijken of er ook iets anders te doen is.

Ik neem voorts, op financiële en organisatorische gronden aan, dat Jansen wel beschikt over een opdrachtenteller, maar niet over een levend geheugen, waarin een link kan worden opgeborgen. Interruptie in de ware zin des woords treden dus niet op. Omdat ik verder zo min mogelijk levend geheugen in Jansen wil veronderstellen; wordt de programmering voor Jansen een heel gebonden kunst.

~~XXXXX~~ Jansen kan tijdens een actie natuurlijk altijd het levend geheugen van de X8 raadplegen (zie 1.2. opm. 1). Ik stel echter, dat frequente raadpleging van het X8-geheugen in het scanproces, wat Jansen nu gaat doen, ongewenst is. Zolang Jansen wacht op werk, moet het geheugen van de X8 ongemoeid blijven.

Het doel van deze notitie is, om er achter te komen, hoeveel "levend geheugen" bv. op flip-flops Jansen moet hebben, opdat uitmaken, wat de volgende actie zal zijn,

een interne aangelegenheid van Jansen zal zijn. Mijn verwachting is, dat de programmering voor de bandlezer als voorbeeld, de noodzaak voor luisterbits zal illustreren.

Ik ga nu, voorlopig met veronachtzaming van OKF ruwweg twee programmatjes voor Jansen maken. Bedenk hierbij, dat $M[f2]$ en $M[f3]$ -zie 1.1.- X8-centrisch Acflop en Inflop zijn, dat IF en VB Jansencentrisch Inflop en Acflop zijn.

De twee programma's zijn:

```
a) "L:P(M| f2|); comment wacht op Acflop van de X8;
    V(VB); comment start bandlezer;
    L1:P(IF); lees B, berg en tel; if telling  $\neq$  0 then
        begin V(VB); goto L1 end;      XXXX
    M| f0| := 2 * f1 + 2 - M| f0|; comment opsteppen van leegwijzer;
    V(M| f3|); comment maak ingreep voor de X8; goto L"
```

en

```
b) "L :P(M| f2|);
    L1:P(IF); lees B, berg en tel; V(VB);
    if telling  $\neq$  0 then goto L1;
    M| f0| := 2 * f1 + 2 - M| f0|; goto L" .
```

Programma a) veronderstelt, dat de "nuchtere toestand" van de bandlezer is met lege B; programma b) veronderstelt dan het B-register vol. Hier zit weer zo'n willekeur als in EWD62 gesignaleerd is. Zonder veel overtuiging kies ik voor programma a); ik kom hier straks nog op terug.

Nu wil ik echter tot uiting laten komen, dat Jansen geen echte ingreep heeft. Dit komt in de programmatjes daarin tot uiting, dat ik ze niet als cycli, maar als gestrekte stukjes schrijven kan, ingeleid door een P-operatie en afgesloten door een "klaar", dwz. de sprong naar de distributeur, de scannerij, met vergeting van het feit, dat we op de plaats "klaar" een proces weer onderbroken hebben. In het inwendige van zo'n stukje mag dan geen P-operatie meer voorkomen.

Versie a) valt dan uiteen in twee losse stukken, waarvan de gewenste afwisseling door een luisterbit LV geregeld kan worden:

```
"P(M| f2|, LV); V(VB); klaar" en
"P(IF); lees B, berg en tel; if telling  $\neq$  0 then
    V(VB) else
    begin M| f0| := 2 * f1 + 2 - M| f0|; V(M| f3|, LV) end;
klaar"
```

De laatste ~~XXXXXXXX~~ V-operatie maakt een ingreep voor de X8 en signaleert via LV, dat wat Jansen betreft, de volgende startopdracht geaccepteerd kan worden.

Ik kan me dit voorstellen, mits op de plug van de bandlezer voor IF en VB drie "draadjes" zitten, een uitleesdraadje voor IF, een false-zet-draadje voor IF en een true-zetdraadje voor VB.

In deze conceptie mogen de laatste twee draadjes niet samenvallen. Doen ze dat wel, dan moeten we een programma maken in stijl b) en dat kost ons dan in Jansen een tweede luistervergunning, nl. om de terugmelding van bandlezer naar Jansen "er is weer een volgend karakter" in geval van nog niet ontstane interesse tijdelijk te kunnen negeren. Nl.:

```
"P(M| f2|, LV1); XXXXXXXXXX V(LV2); klaar" en
"P(IF, LV2); lees B, berg en tel; V(VB); if telling  $\neq$  0 then V(LV2) else
    begin M| f0| := 2 * f1 + 2 - M| f0|; V(M| f3|, LV1) end; klaar"
```

Vergelijking van deze twee programmaatjes illustreert aardig, hoe we de flip-flop, die we bij de bandlezer kunnen uitsparen, er bij Jansen weer verkrijgen.

We veronderstellen op het ogenblik dus een mechanisme, dat de uitkomsten van logische producten scant en op grond van het = 1 zijn naar een bepaald programmatje springt. Ik neem aan, dat daar de P-operatie voltooid wordt. In het laatste programma kan ik daar $IF := \emptyset$ uitvoeren (identiek aan $V(VB)$). In plaats van meteen ook $LV2 := 0$ (pardon, ik bedoel $LV2 := \text{false}$) uit te voeren, kan je de luisterbit laten staan en hem pas afzetten, als de telling op blijkt.

Nu het programma in versie a) met inbegrip van de OK-flipflop. Ik veronderstel dat er nu op de plug een uitleesdraadje voor OKF zit en een true-zetdraadje voor HOK. Om schrijfwerk (dwz. programmeruimte) te besparen, lopen de twee stukjes na enige tijd samen.

```
"P(M| f2|, LV);
  if herstel OK-toestand in startmagazijn then begin HOK:= true; V(VB); klaar end
                                     else
      if OKF then begin V(VB); klaar end
                                     else
L+ begin meldt OKF in klassewoord;
    M| f0| := 2 * f1 + 2 - M| f0|; V(M| f3|, LV); klaar end "
```

en

```
"P(IF); if herstel OK-toestand then goto L;
      lees B, berg en tel;
      if non OKF or telling = 0 then goto L;
      V(VB); klaar".
```

Wij merken hierbij op:

- 1: dat 1.2. opm.1. slaat op de test "if herstel OK-toestand" in de eerste regel van het tweede programma
- 2: dat een analoog programma in versie b) ook vast wel te maken is
- 3+ dat het symbolisch afwerken in het geval van een groot startmagazijn niet Jansen gedurende lange tijd blokkeert
- 4+ dat we "HOK:= true" op zichzelf al als start van de bandlezer kunnen beschouwen.
- 5: dat we een IF1 en IF 2 hadden kunnen invoeren, waarbij IF1:= true een nieuw karakter aankondigt en IF2:= true een voltooide herstelling van de OK-toestand, wat ik echter niet gedaan heb, om het aantal bits, dat gescand moet worden in de toestand "klaar" niet nodeloos te laten oplopen. Ik laat voorlopig nog in het midden, hoe deze scan plaatsvindt: als deze scan als cyclus uitgeprogrammeerd wordt, dan hangt de tijd die verstrijken kan tussen het true worden van een flip-flop (of logisch product) en het ogenblik, dat Jansen dit ontdekt, ook als Jansen niets anders te doen heeft, van de grootte van deze cyclus af.

In het geval van uitgeprogrammeerd scannen kunnen we "klaar" nog op minstens twee manieren programmeren: of als een sprong naar een vast punt, waar we de boolean condities op een vaste volgorde gaan onderzoeken, en als we niets vinden, dat gedaan kan worden, springen we weer naar het begin van deze lijst terug. Of: we onderzoeken de boolean voorwaarden cyclisch, maar als we een programmatje uitgevoerd hebben springen we midden in de scanketen, vlak onder de plaats, waar we hem verlaten hebben. Mengvormen zijn ook nog mogelijk, maar dan lijkt het me al ras heel moeilijk, om iets over maximale wachttijden etc. te weten te komen.

Ik heb wel idee, hoe ik hier enige gegevens over zou kunnen krijgen, of ik er binnenkort aan toe kom, is een tweede. Totzover lijkt me de moraal, dat het verwijderden van de plicht om link's te kunnen bergen tegen de prijs van extra luisterbits me een voordelig handeltje lijkt.