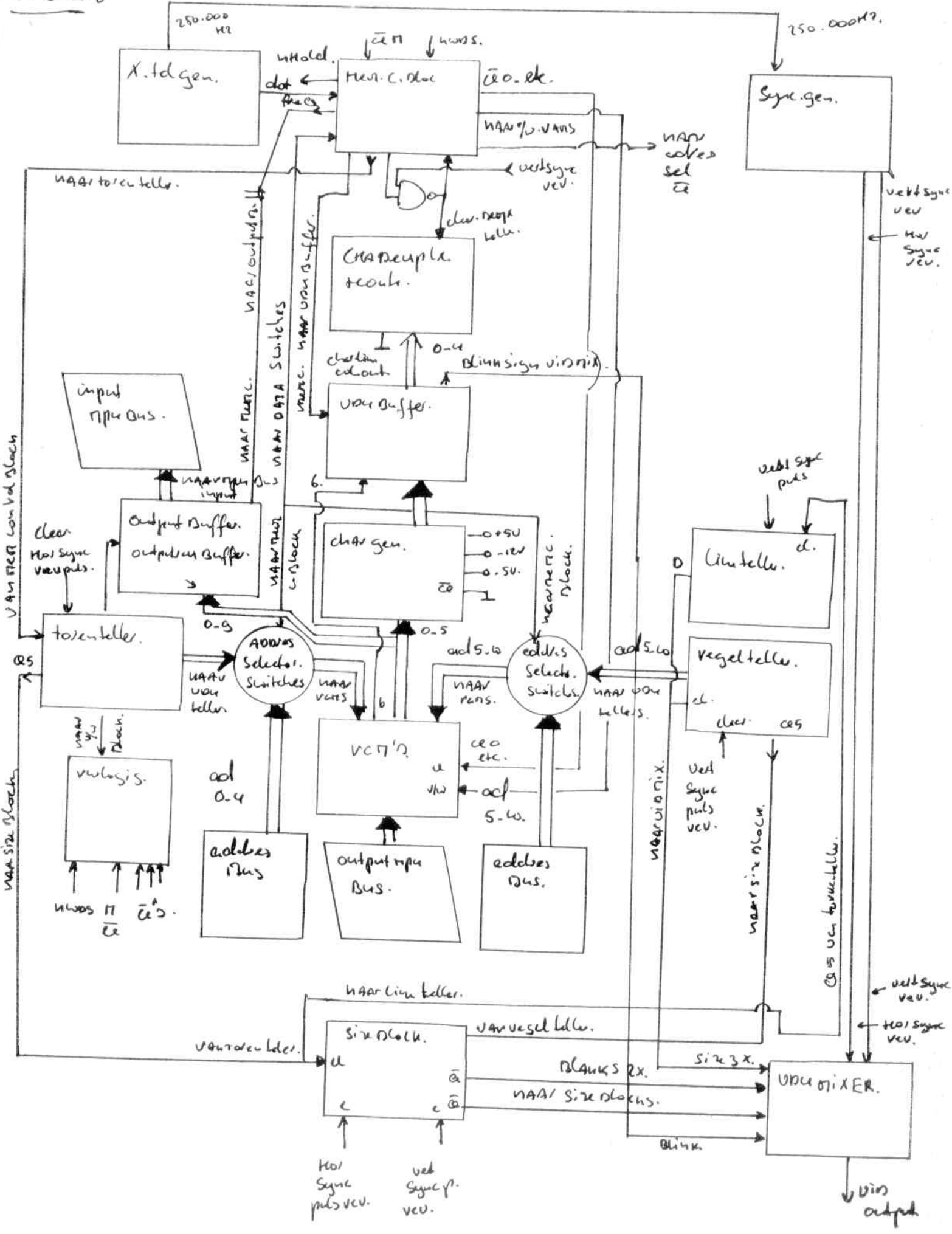


VIOWO



Beelt opbouw.

Deze schakeling ~~xxxx~~ zorgt voor het refreschen van een willekeurige hoeveelheid geheugen. Tevens wordt een grootgedeelte van de zelfde schakeling gebruikt als vdu. Bijde processen lopen geheel door elkaar heen en de gebruiker merkt hier niets van.

Opbouw van het beeld.

Het beeld bestaat uit regels dat zijn horizontale groepen karakters. De vertikale groepen karakters heten torens.

Opbouw van een kearakter.

Elk karakter bestaat uit een horizontaal aantal lijnen lines genoemd en tien in totaal. Hiervan worden er 7 voor het feitelijke karakter gebruikt en 3 als spatie lijnen. Elk karakter bestaat vertikaal uit 6 kolomen waarvan er 5 voor het karakter en een als spatie kolom wordt gebruikt.

```

00000x00000x
00000x00000x
00000x00000x
00000x00000x
00000x00000x
00000x00000x
00000x00000x
000000x00000x
xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx- spatie lijnen.
xxxxxxxxxxxxxxxxxxxx
      x      x
      x      x - spatie kolomen

```

Memory control block+ demultiplexer.

Iedere karakter line bestaat uit 6 kolomen De frequentie van elke punt is de ~~21x~~ Dot frequentie. Zo'n karakter line ontstaat uit een demulti-plexer die de line output van een karakter generator uitleest. De uitgang van een karakter generator bestaat uit 5 kolomen deze vormen de kolommen ingangen 0-4 van de demulti-plexer, de 5de ingang is aan massa gelecht maar wordt wel uitgelezen zo ontstaat er een spatie kolom tussen de karakters. De teller die de demulti-plexer stuurt wordt geklokt door het memory control block en de teller kan zowel door het horizontale syngronisatie signaal (voor aan het begin van een regel) als door het Mem.control block gerest worden. Boven de Demulti-plexer staat het memory control block . dit onder deel bestaat uit een binaire teller + een 4 naar 16 kanaal decoder. de uitgangen van de teller zijn met de ingangen van de decoder verbonden. De teller wordt geklokt door een signaal van 2maal de dot frequentie. uitgang a ven de teller stuurt de teller van de demulti-plexer. De mem c bl. teller kan dus als een karakter uit zes kolomen bestaat maar tot 12 tellen (0-11) Het clear sig naal wordt dan ook afgeleid van ~~het~~ uitgang 12 van de decoder. Als deze uitgang nul wordt zullen alle twee detellers ogenblikkelijk worden gereset als een gevolg hiervan is deze uitgang maar enkele ns laag . Bijde teller. kunnen ook zoals al gezecht is gereset worden door door het horizontale syngronisatie signaal.

Programmering van het Memory controlblock

Het mem. c. blk. doorloopt 1 cyclus gedurende elke lijn van 1 karakter. deze cyclus heeft 2 funkties:1. het regelen van de vdu opzich zelf
2. het regelen van de mpu acces.

stap ce rams

funktie

0	$\bar{ce} 1$ low	lees de uitgangen van de karakter generator in de karakter line buffer.
1	afh van mpu addr.	1. als nhold is low set ff 1 2. switch ram to mpu.
2	idem	deduren deze stap wordt er gewacht om de min. acces time te overbruggen.
3	idem	hier wordt een puls gegenereerd die de inhoud van de betram lokatie in de output buffer leest.
4	idem	Als nwds=0 + nhold =0 produceer dan een r/w puls voor de dyn . mem. Hierdoor wordt de inhoud van de databus in de de betreffende ram lokatie gelezen.
5	idem	xxxxxxxixrnxgxnaxrnxvdux 2. wait.
6	idem	1. ram terug naar vdu 2. als ff 1 is geset reset ffl + nhold buffer=ff1
7	all $\bar{ce} = 0$ etc = 1	wait
8	idem	maak r/w low (dit is de refresh puls)
9	idem	wacht en hoog intussen de kolomen teller met een op
10	alleen $\bar{ce} 1$ low	wait
11	idem	wait
12	-----	reset zowel mem.c.blk.teller als demultiplex teller.

Memcontrol stap bespreking.

onafh: Als er een verzoek van de mpu komt voor mem acces over de memreq line dan wordt ff 2 geset ten teken voor de mem. c. block dat de mpu acces vraagt. voornaamste functie van deze flip flop is echter dat hij de mpu via de nhold line in de wacht toestant brengt. de flip flop kan alleen gereset worden als de memmpu cycle is voltooid (zie daarvoor stap6)

stap0: ~~xxxx~~ tijdens deze stap kunnen we aannemen dat:

1-de vdu over de ram bussen beschikt.

2-de ce(rev) van kkaart 1 laag is.

3-toestanden 1+2 reeds enige tijd gelden.

Op de uitgang van de rams staat nu de ascii code van het desbetreffende teken. tijdens deze stap wordt deze informatie in de vdu buffer gelezen. (de informatie van de karakter generator meen ik)

stap1: tijdens deze stap wordt onvoorwaardelijk ff3 geset wat tot gevolg heeft dat de ram bussen over worden gezet op de mpu. (voor het reseten van deze ff zie stap6. Ook wordt als ff2 is geset ffl geset. ffl kan pas worden gereset als de mpu mem cycle is voltooid.

waarom is ffl nodig? ste dat we alleen ff2 zouden hebben en er komt een memreq puls tentijde stap 5. ff2 zou dan bij 6 worden gereset zonder dat de gehele mpumem cycle is voltooid.

stap2: ~~xxxx~~ Deze stap heeft alleen tot doel de gewenste vertraging voor de trge geheugens te geven die naar verhouding een lange acces time hebben.

stap 3: ~~xxxx~~ De gegevens uit de door de mpu geselecteerde locatie worden nu in de output buffer gelezen. Dit gebeurt altijd onafhankelijk van of het nu een read of een write cyclus is.

als ffl of ff2 niet geset zijn vint toch deze puls plaats het kan immers geen kwaad geheugens uit te lezen.

stap4: als aan de hieronderstaande voorwaarde n is voldaan vint er via een and gate een r/w pus plaats.

1. nwd=0 (dan hebben we met een write cyclus te maken)
2. stap 4 aan de beurt is dwz=0 (nogal logisch)
3. ffl is geset .

deze logisch funktie ontstaat uit twee two input ors die geschakeld zijn als een three input or.

stap5: zie stap 2

stap6: tydens deze stap wordt ff3 gereset (dit kan ook gebeuren tydens elke teller clear, zodat de vdu altijd de beschikking over de ram bussen heeft by stap nul). verder worden de flip floppen 1 en 2 gereset als tenminste ffl was geset . Deze voorwaarde is belangrijk omdat we als we ons hier aanhouden weten dat de mpu mem cycle voltooid is.

stap7: alle ram cs(rev) ingangen van de ram geheugens worden nu hoog)

stap8: via de al in 4 genoemde ~~ramxp~~ and port wordt er nu weer een r/w puls gemaakt die de refresh puls is

stap 9: zie stap 7 + tevens wordt de toren teller met 1 verhoogt.

stap 10 nu worden alle ce(rev) hoog gemaakt behalve no 0 zodat de vdu op de goede geheugen pagina staat. verder wachten.

stap11: idem

stap12: resetl. mem.c.teller

2. demultiplexer teller.

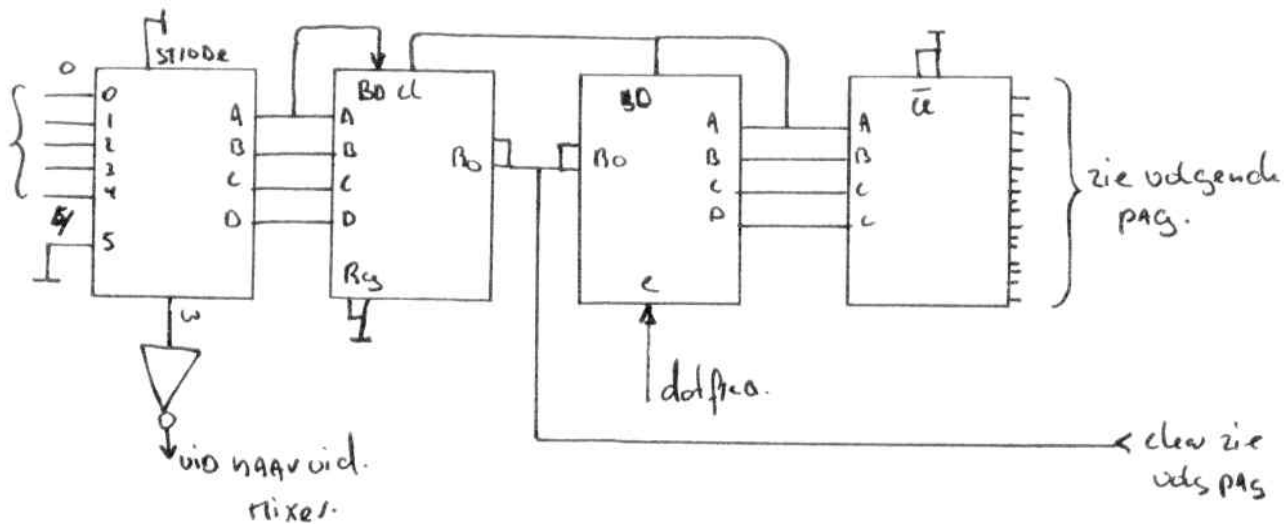
74150

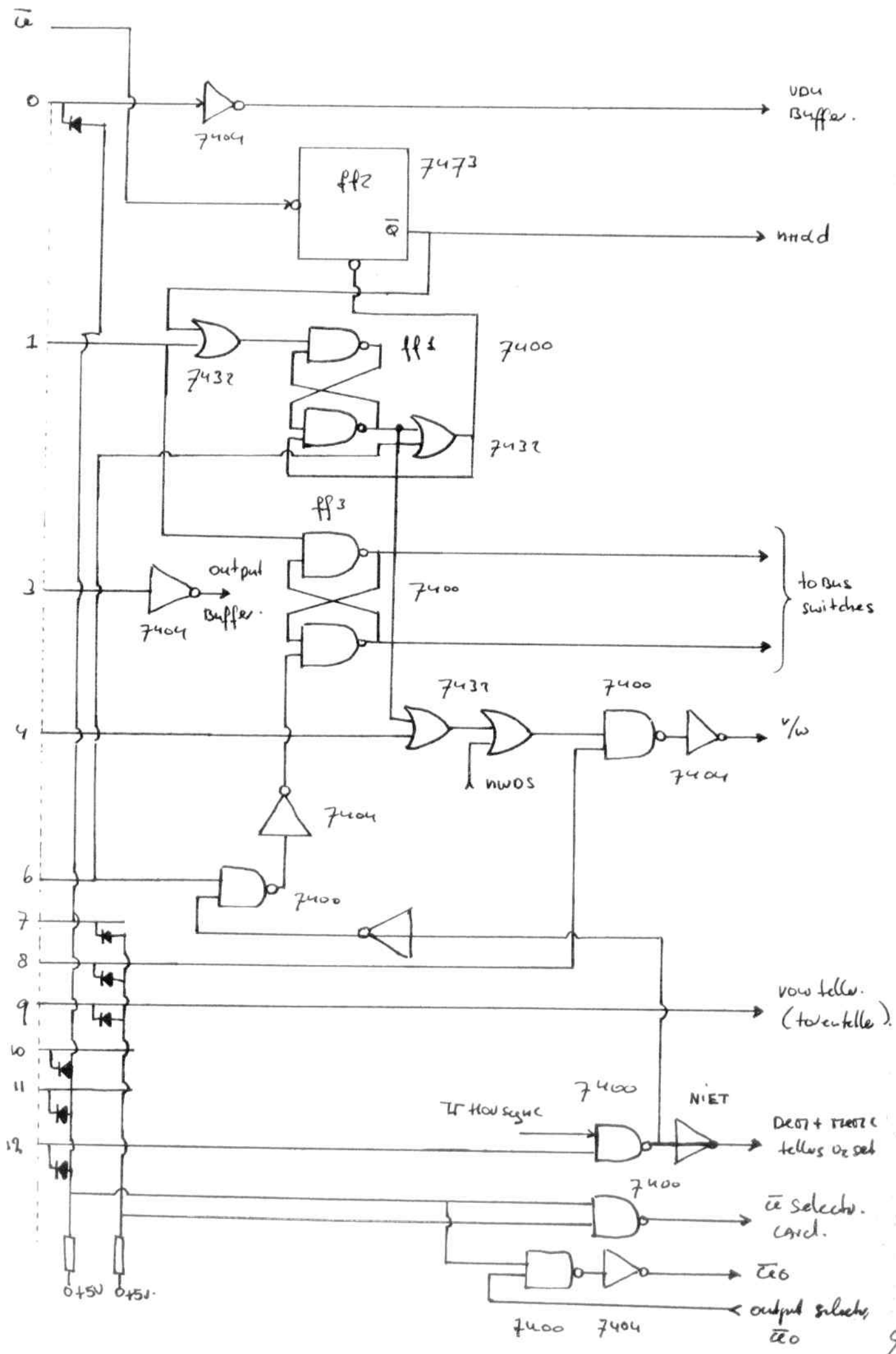
7490

7493

74154

NAAR
VON
Differ.

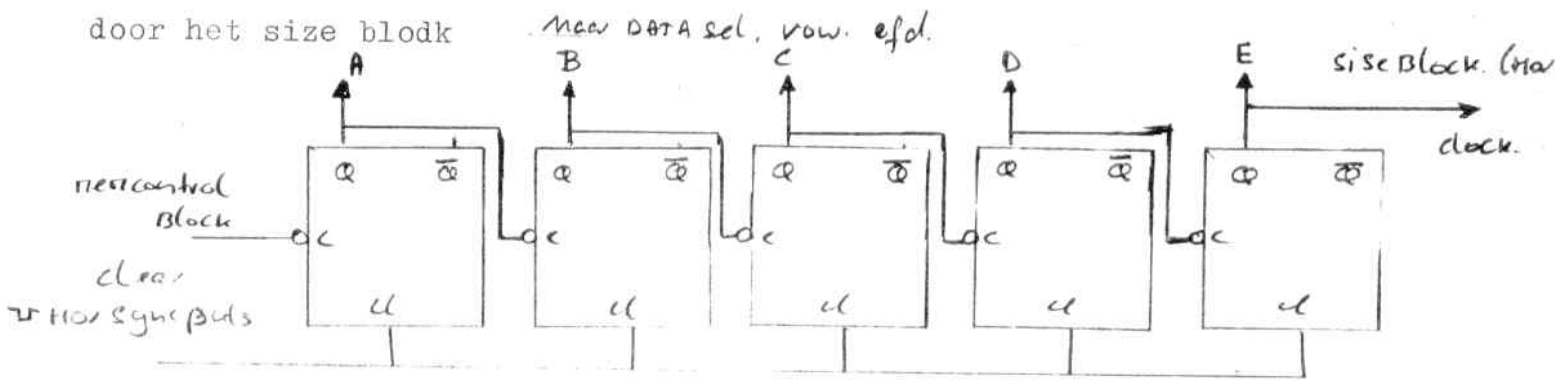




De toren teller

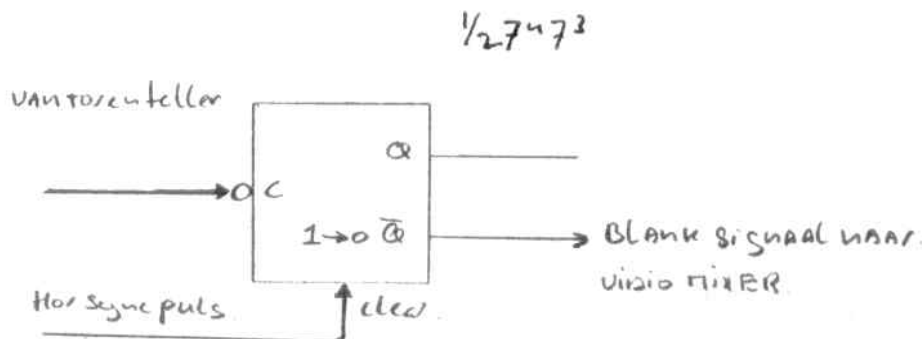
Deze teller is in feite ook de refresh counter.

De clock in gang van deze teller wordt gestuurd door een puls uit het memcontrol block. Deze puls treed al op tijdens het displayen van de kolomen van een lijn van een karakter. Dit is omde traagheid van de geheugens teover bruggen. De teller wordt aan het begin van iedere regel gereset door de horizontale syngronisatie puls. dit zorgt er voor dat alle torens netjes onder elkaar staan. Tijdens de hor.sync.puls is geen mpu acces mogelijk. Let op dat de vertikale syngronisatie puls geen enkele invloed heeft op de reseting van de toren teller. Dit houd in dat het tellen van toren teller niet te stoppen is . Dit is noodzakelijk voor een korte mpu acces time voor de mpu. Het aantal kerakters op een regel wordt beperkt door het size block



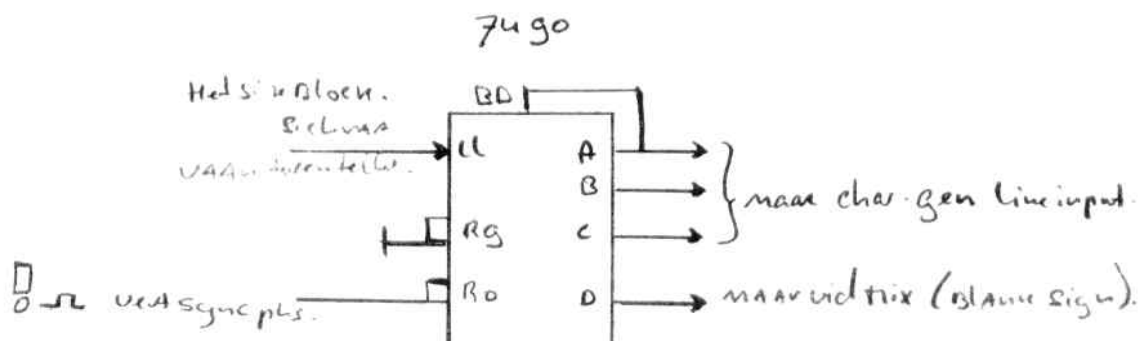
Size block. (horizontaal)

Bestaat uit twee JK flip flops die geklokt worden door een van de adres lijnen. van de toren teller . dit heeft tot gevolg dat de flip flop omklapt de rest van de lijn blijft nu donker. de flip flop wordt gereset door de hor.sync.puls.



De Line Teller.

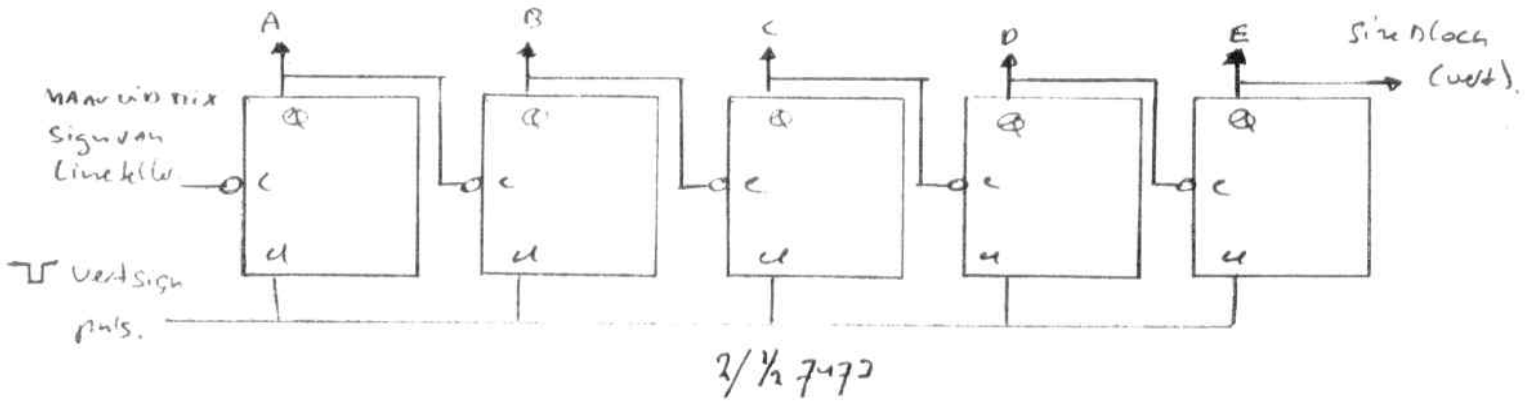
-----Deze teller telt de lynen waaruit een regel met karakters bestaat. De uitgang van deze teller is aangesloten op de lijn ingang van de karakter generator. Het kloek signaal voor deze teller wordt geleverd door de toren teller. en is het zelfde signaal dat door de toren tellers aan het size block wordt geleverd. Dit signaal wordt namelijk geleverd als de machine klaar is met het vertonen van de laatste kolom van een regel. Het clear signaal is het vertikale syngronisatie signaal. Omdat de teller een tien teller is en een karakter maar uit 7 lijnen bestaat gebruiken we 3 spatie lijnen. Een van deze lijnen is ingeprogrameerd in de karakter generator zodat een karakter uit de karakter generator in feite uit 8 lijnen bestaat. voor de overrige 2 lijnen moet een blank signaal opgewekt worden. dit wordt geleverd door uitgang d als deze lijst er beeld onderdrukking plaats.



De Regel Teller

-----Deze teller telt de regels. Het kloek signaal voor deze teller wordt geleverd door het video blank signaal van de line teller. Het clear signaal voor deze teller wordt verzorgt door de vertikale syngronisatie signaal. Hier door wordt de teller gereset aan het begin van elk nieuw beeld. Een van de 5 uitgangen van deze teller levert het signaal vaar het size block (vertikaal)

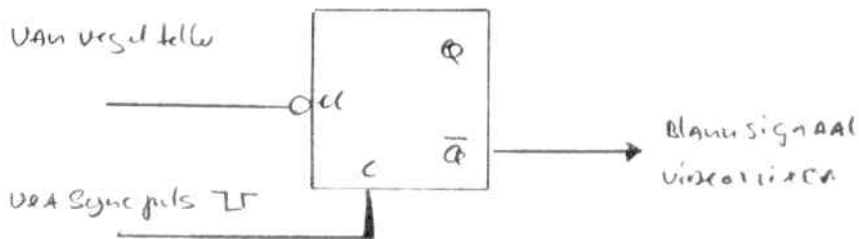
naam adres van de bus (adres).



size block (vertikaal)

bestaat uit een JK flip flop die geklokt wordt door het hierboven beschreven signaal dat eronder dat de laatste karakter van ~~de~~ het beeld vertoond is. de flip flop wordt gereset door de verticale syngronisatie signalen. de uitgang van de flip flop levert de nodige blank signaal.

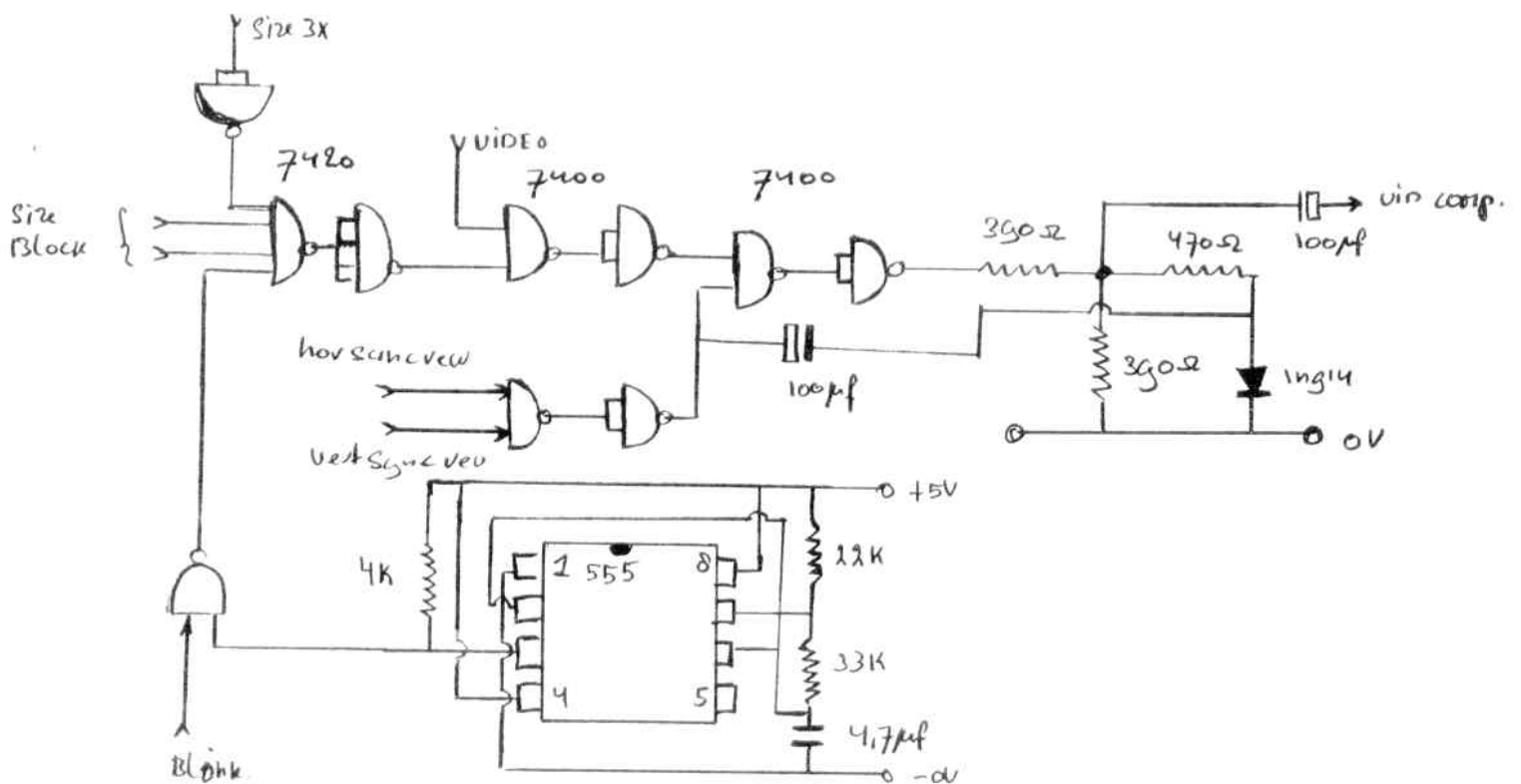
1/2 7472



Video Mixer Block.

Dit block zorgt voor de juiste mixing van :

- a. De bijde synchronisatie signalen(hor sync,rev. + vert sync.rev)
- b. Video signaal (1=wit , 0= donker)
- c. De video blank signalen1.drie lijn onder drukking (1= onderdrukking)
 - 2.size block signalen. (0=beeld onderdrukking)
 - 3.flash signalen.(0=norm, 1=flash)

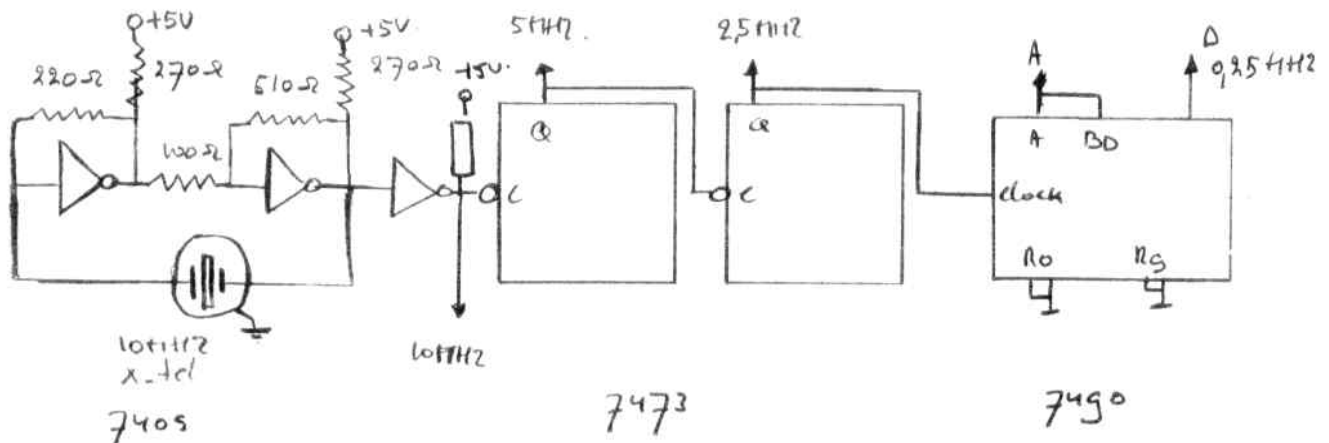


Master klok generator.

De master klok generator bestaat uit een kristal generator die oscilleert op een frequentie van 10.000.000 Hz. Van deze worden ten eerste twee andere signalen afgeleid 1. een signaal van 5.000.000 Hz

2. een signaal van 2,500.000 Hz

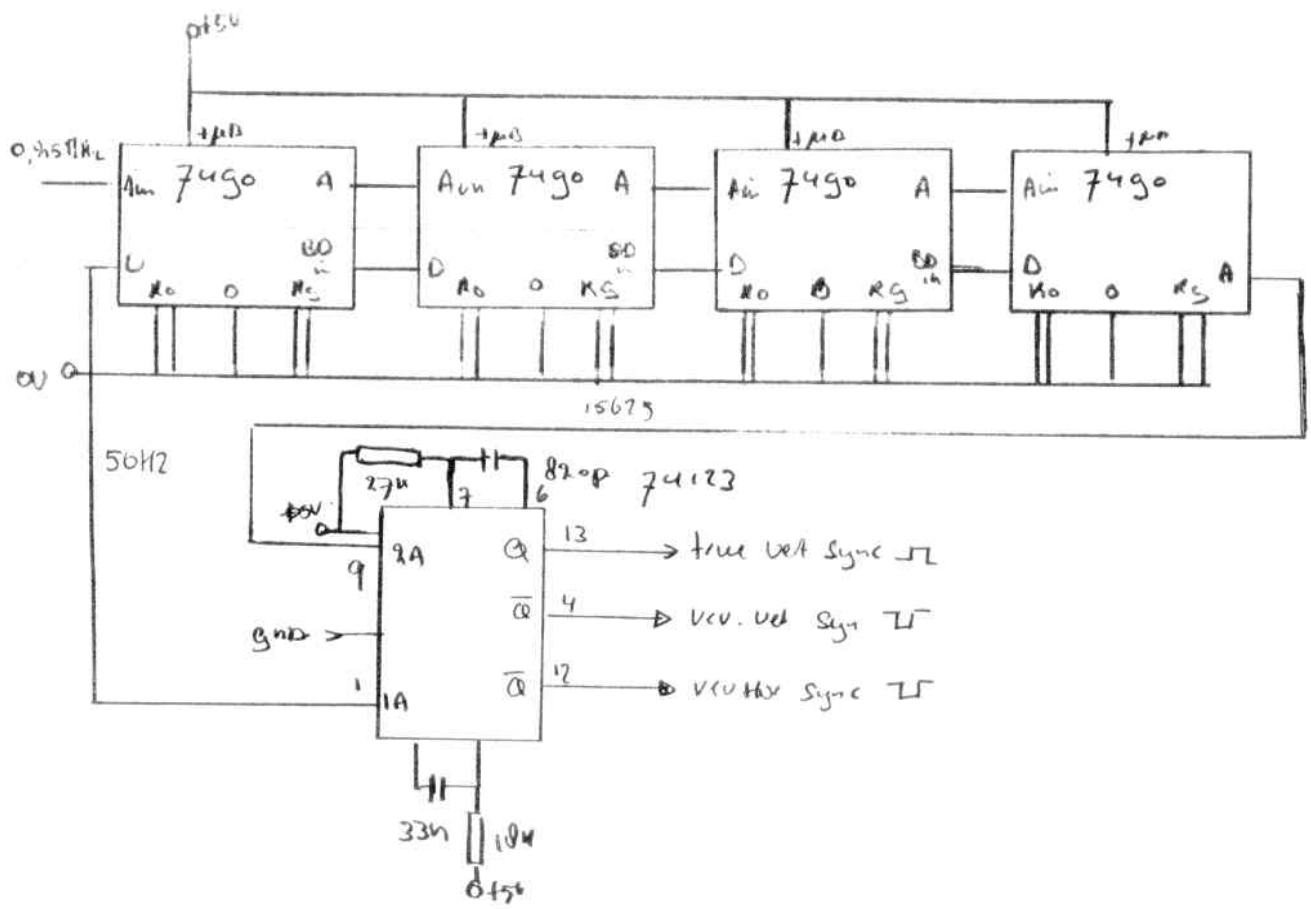
de signalen 10 MHz 5MHz en 2,5 MHz kunnen gebruikt worden als dotfrequenties. het signaal van 2,5 MHz wordt ook nog door tien gedeelt zodat een signaal met een frequentie van 250.000 Hz ontstaat ten behoeve van de sync. signaal generator.



De sync signaal generator.

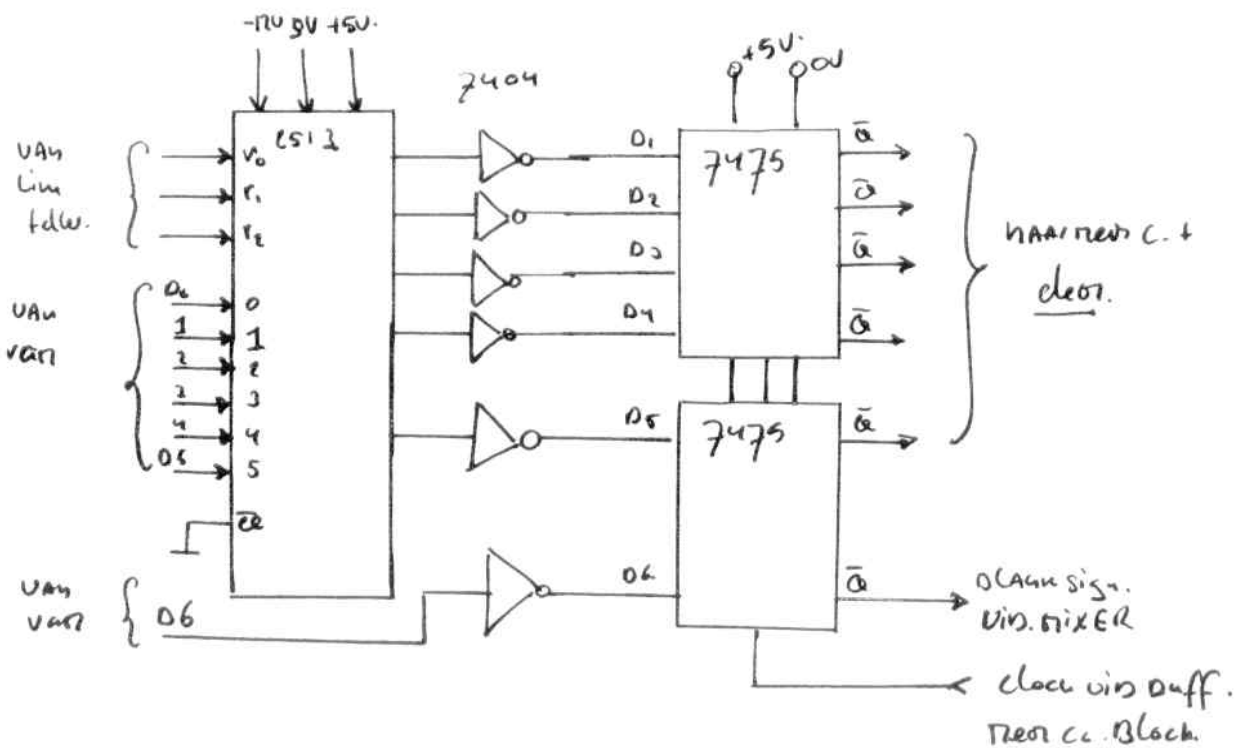
De sync. signaal generator bestaat uit een deler trein en een puls vormer. Het signaal van 0,25MHz wordt eerst door 8 gedeelt door 3 twee delers er ontstaat dan een signaal van 31250 Hz dit signaal wordt dan enerzijds door twee gedeelt zodat het horizontale synchronisatie signaal ontstaat van 15625 Hz anderzijds wordt dit signaal door 625 gedeelt door 4 5delers zodat er het verticale synchronisatie signaal van 50 Hz ontstaat. Beide synchronisatie signalen worden dan toegevoerd aan een puls vormer die voor het horizontale synchronisatie signaal een puls van 4us en voor het verticale synchronisatie signaal een puls van 250 us beide signalen zijn in zowel ware als in geïnverteerde vorm beschikbaar.

De ware synchonisatie signaal (vertikaal) wordt gebruikt voor het reseten van de line teller die daar toe een positief (van 0 naar 1) signaal behoeft , De byde omgeke rde signalen worden gebruikt voor : het reseten vande toren teller, de regel teller , de char dempx, de memcontrol block, de size blokken (hor en vert). verder worden deze signalen gebruikt als synchonisatie signalen naar de video mixer.



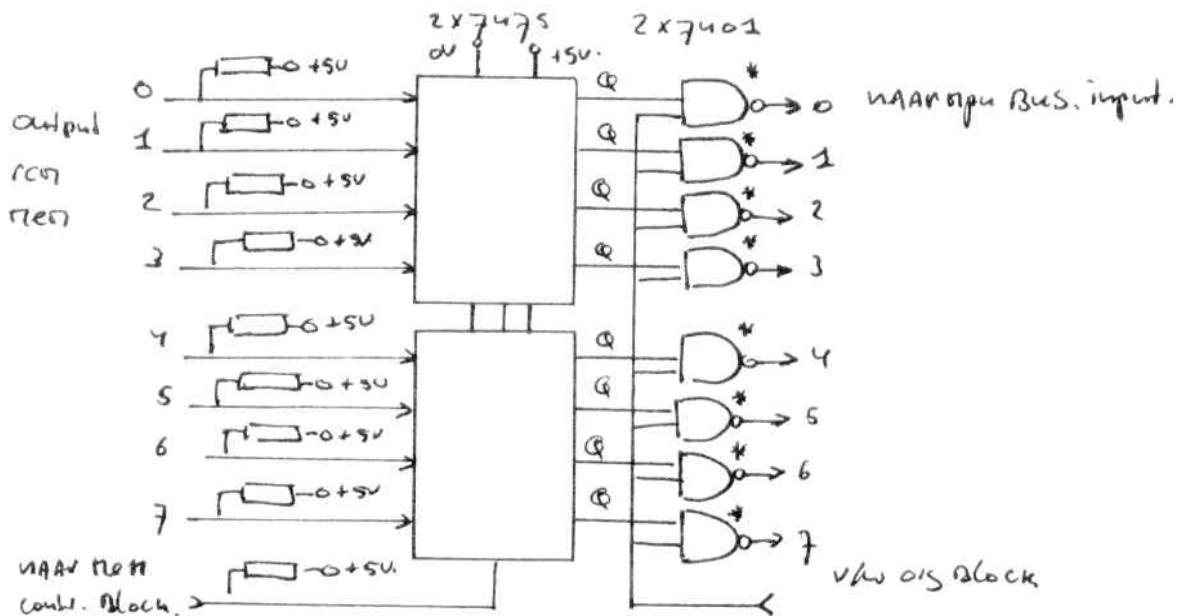
Char generator + vdu buffer.

De uitgangen van het ram geheugen zijn niet alleen op de output buffer maar ook op de karakter generator aangesloten. De karakter generator heeft 6 ingangen waarmee je een karakter kunt kiezen. In totaal zijn er dus 64 karakters mogelijk. Voorts zijn er nog drie ingangen waarmee je de karakter lijn kunt kiezen. Omdat de karakter generator een cmos i'c is en de buffers een fan in van twee hebben moeten er in de karakter output lijnen buffers opgenomen worden. Dit houdt wel in dat van de buffer de reverse uitgangen moeten worden gebruikt. Omdat de karakter generator maar 6 ingangen heeft en de ram 8 uitgangen blijven er twee over. een van bijde over gebleven uitgangen wordt rechtstreeks met een van de vdu beffers verbonden dit signaal moet later dienen als blink signaal. De buffer wordt geklokt door het memory control block. De karakter generator heeft nog een chip enable ingang deze is met massa verbonden.



Output buffer.

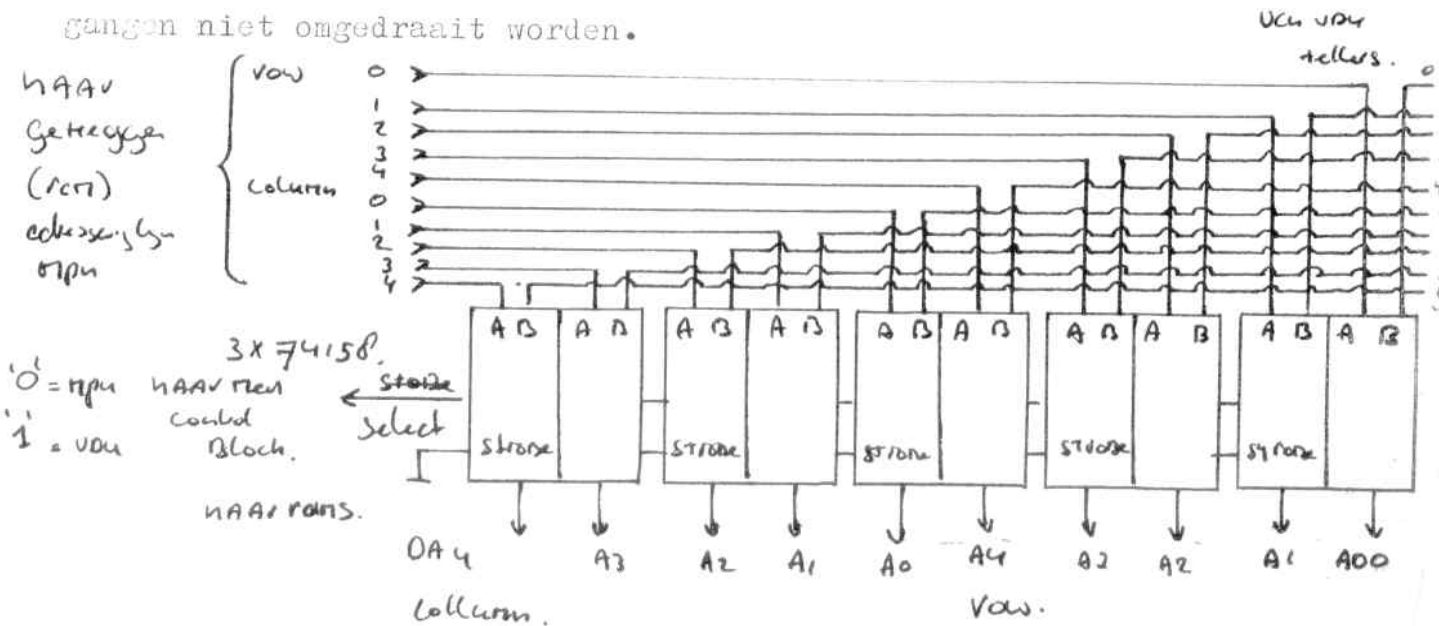
De output van geheugen naar mpu gebeurt via een 8 bits latch en 8 wired or poorten. De latches worden gestuurd door het memory control block. De wired or poorten worden gestuurd door een schakeling die alleen toestemming om de poorten aanzet geeft als en ffl is geset en nwd=0. Bovendien moet de ingang van de gdic kaart gestuurd worden



* = open collector.

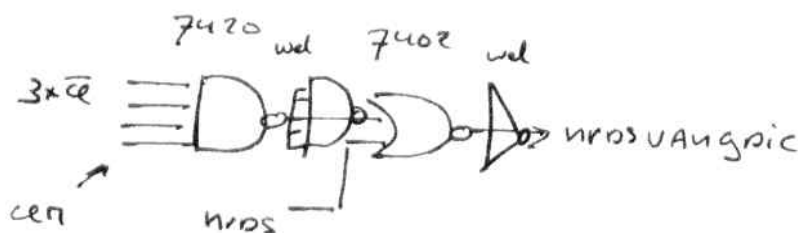
De addresses switches

Deze schakelen de adres bussen van de ram om van vdu naar mpu. Hij staat volledig onder controle van het memory control block. Let op dat de ingangen niet omgedraait worden.



R/W Block

Bied plaats aan 4 enable ingangen dit zijn 4 ingangen die aangesloten kunnen worden op general memory pointer uitgangen. een ingang is aangesloten op cem zo gauw dus een van de ram geheugens aangesproken worden kan die inwerking treden als $nwds=0$ tenminste.



CEM = ram kaart aanzetter.

als de ce van de 74154 die gebruikt wordt voor de ramkaarten en verbonden is met ad 14 laag wordt en $nwds+nwds=0$ dan wordt cem 0. Hierdoor treedt het ram mechanisme in werking en wordt nhold low.

