

Z-MODEM MET 32-BYTES START SEQUENCE OP EEN 1200/75 BAUD MODEM MET INTERSPEEDER NU MOGELIJK.

INLEIDING

FIFO buffer voor elke speedconverter met standaard UART.

Zoals:	AY - 3 - 1014		+5 tot +12 V
	AY - 3 - 1015D		+5 V
	AY - 5 - 1013]	
	COM 2017]	+5 én -12 V
	TR 1602]	
	CDP 6402		+5

Aangeboden: twee versies!

De ene met een buffergrootte van 16 bytes op een mini printje of een grotere print met 16, 32 of 48 bytes buffer. De truuk zit hem in de goedkope (eindelijk) CD 40105B FIFO IC's.

(Enkele jaren terug ben ik opgehouden met ontwerpen omdat vergelijkbare IC's of betere bijv. van TI (74224, 7030, 3351 of 4703) toen minstens f 50,-- per stuk kosten en tevens enkele honderden mA's konsumeerden. Nu in C-MOS laag energie verbruik, lage prijs en zelf doorklokkend, dus zelfs geen klokpuls meer nodig.

Nu wachten tot de 74HCT7030 voor de hobbyist betaalbaar is.)

De FIFO uitbreiding bestaat uit een klein hulpprintje met 40-polige flatcable (of DIP) stekker. Die gaat in de originele UART voet. UART in de voet op het hulpprintje en de ombouw is klaar!

EEN VOOR-VERHAAL

Een goede reden om een 1200/75 baud modem te gebruiken is bijv. omdat het goedkoop is of omdat je dat nu eenmaal nog hebt liggen. Om het op een eenvoudige manier te kunnen gebruiken met bijv. pcplus op de PC moet het voorzien zijn (of worden) van een speedkonverter met een UART. Enkele modems hebben die reeds in zich (Telektor september 1984), andere modems kunnen er mee uitgerust worden. Zelfbouw ontwerpen zijn reeds gepubliceerd in Elektuur en zelfs in het HCC blad. Er is maar één groot probleem met deze konstruktie: nl. als er meerdere bytes snel achter elkaar aangeboden worden. Dat gebeurt al bij heel snel intikken (merken we niet allemaal zo vlug), of bij het aanbieden van bytes via een algoritme (ASP file in pcplus), of een download protocol gebruikt een start sequence van meer dan één byte (Z-modem zelfs 32 bytes), of een checksum van meer dan één byte bij ieder verzonden blok. Bij elk van de hierboven opgenoemde gevallen gaat het mis of goed mis!!

De reden? Voordat de interspeeder klaar is met het omzetten van één byte van 1200 naar 75 baud kunnen er al minstens $1200/75-1=15$ bytes verloren gegaan zijn! Wat we nodig hebben is een tussengeheugen dat snel deze bytes opvangt bij het Rx gedeelte van de UART en ze op korrekte wijze, zonder er één kwijt te raken, in een gepast ritme weer aanbiedt aan het Tx gedeelte. We noemen zo'n buffer een first in / first out buffer of kortweg FIFO. De grootte van het geheugen is afhankelijk van wat we er mee willen; uploaden blijft zinloos in 75 baud. (Schakel dan over naar 300 baud als je niks anders hebt, of met komputervrienden onderling, schakel dan over op reverse 75/1200 mode i.p.v. 1200/75.) Voor gewoon dagelijks gebruik (kommunikatie met PCPLUS + ASP files, xmodem, telink, zelfs kermit) is 16 bytes voldoende gebleken. Wil je ook Z-modem dan is 48 bytes een juiste keus. Anders werkt het gewoon niet. (Zmodem schakelt dan wel terug naar een simpeler protocol en gaat verder!) Eén rijtje FIFO's weglaten en input / output doorverbinden geeft voor zuunige mensen (ons Zeeuwen.....) een tussen oplossing met 32 bytes i.p.v. 48 bytes op dezelfde print.

ONTWERPGEDACHTEN

Omdat in een eenvoudige RS 232 verbinding hardware handshake op byte nivo niet makkelijk en handig is, juist daarom hebben we een FIFO nodig, is in de ingang van het geheugen (de Rx kant van de UART) ook geen poging gedaan dit aan te brengen. De FIFO buffer moet gewoon ieder karakter dat aangeboden wordt opnemen en snel doorschuiven naar de Tx kant. Dat laatste gebeurt in de toegepaste IC's gelukkig volledig automatisch met behulp van een zelfoscillerende inwendige clockpuls.

Tussen de lage snelheid Tx kant van de UART en de uitgang van het geheugen moet wel korrekte handshake op karakter (byte) nivo plaatsvinden. Dat gebeurt d.m.v. twee multivibrator schakelingen, de timing pulsen uit de UART en de toch al aanwezige Tx clock van de UART (=16x de baudrate), scheelt weer een 555 IC. Om meervoudig "uitclocken" van dezelfde byte te voorkomen, moet het hele handshake spel (timing MMV's) zich afspelen binnen één periode van de Tx clock. Bij de volgende opgaande flank van de Tx clock is dan een nieuwe stabiele situatie aanwezig!

OVER DE SCHAKELING EN DE SCHAKELWIJZE

Resetpuls

Om er voor te zorgen dat alle geheugens, buffers en registers leeg zijn na het inschakelen, is voorzien in een nette resetpuls van +/- 1 seconde na de opstart van de +5 volt. Deze puls dient tevens als vervanging voor de resetpuls voor de UART, indien aanwezig!

Het ontvangen van een byte

Bij 1200 baud input is de tijd tussen twee bytes $1/1200 = 833$ usec. De kortste timing pulsen in de UART in het Rx deel liggen in de grootte van $1/19200$ (16x baudrate) = 52 usec. De tijd om een byte goed in de FIFO te krijgen is +/- 0,5 usec. Tijd genoeg dus. Voor de input strobe is gekozen voor een delay (met R.C) van +/- 20 usec. Lekker lang, maar toch nog korter dan één klokpuls, geen verwarring mogelijk.

Wanneer er een byte gereed staat in de Rx buffer geeft de Data Ready (pen 19) een positieve strobe puls af. Op de voorflank wordt de byte aangeboden aan de FIFO buffer. Na enige tijd is dit geaccepteerd door de FIFO (dit gaat dus zeer snel) en omdat verdere handshake toch zinloos is, wordt automatisch na een tijd, bepaald door R.C, d.m.v. een negatieve puls op pen 18 (Data Ready Reset) de strobe puls gereset. Data Input Ready (DIR, pen 2) van de FIFO dus niet aangesloten. Hoewel de meeste UART ingangen voorzien zijn van pull-up stroombronnen en TTL/C-MOS compatibele uitgangen hebben, was een pull-up R van 3K3 toch nodig voor een betrouwbare werking. Het nivo was OK voor TTL, maar voor de gebruikte C-MOS schmitt-trigger was een "1" net niet hoog genoeg.

Het uit"clocken" naar de UART

Daar de signaal timing van FIFO en UART te snel en kritisch zijn om zomaar aan elkaar te knopen is voorzien in passende outputhandshake. Om een 555 IC uit te sparen is voor de timing de Tx clock van de UART gebruikt, een betere timing met de UART is er niet.

De Tx clock = $75 \times 16 = 1200$ Hz, T = 833 usec, tussen één voor- en achterflank dus ruim 400 usec. De limieten van zowel FIFO als UART liggen bij +/- 0,5 usec. Ergens tussen deze twee uitersten zal het handshake gebeuren altijd werken en zouden we de tijden van de twee MMV pulsjes graag gehad willen hebben. Tussen de 5 en 50 usec is dus prima, zeer snel, maar voldoende duidelijk voor de UART.

Maken we om eens te proberen de MMV C's een paar maal groter, bijv. 10 nF en 4n7 dan werkt het nog steeds.

De werking van de uitgang

We gaan er even van uit dat de Tx buffer leeg is, input buffer Tx ready (pen 22) is dan hoog. Binnen enkele usec nadat er een byte in de FIFO ging, gaat Data Output Ready (DOR, pen 14) omhoog, ten teken dat de betreffende byte reeds op de output staat. Bij de eerstvolgende opgaande flank van de UART Tx clock wordt via N4 en MMV1a een Tx load puls afgegeven aan de UART. De respons hierop is dat de Tx ready (pen 22) laag wordt. Deze blokkeert hiermee de MMV. Op de

achterflank van de load puls gaat de byte intern in de UART verder naar de Tx shift out buffer en tevens geeft MMV1b een korte negatieve puls aan het FIFO buffergeheugen, die hierop weer reageert door DOR laag te maken. De volgende byte, indien aanwezig, verschijnt even later op de output poort en DOR gaat weer omhoog. Was er echter géén volgende byte, dan blijft DOR laag, de clock pulsen hebben dan nog geen effect op MMV1a. Ook moet Tx ready weer hoog zijn, dan pas wordt MMV1a opnieuw vrijgegeven. De output cyclus begint dan weer van voren af aan.

Resumerend:

DOR (Data Output Ready) hoog staat voor: data aanwezig.

DOR laag betekent: geen data of nog niet gereed.

Tx ready blijft laag totdat de Tx inputbuffer zijn data kwijt kan in de Tx shift out buffer en weer een volgende byte kan overnemen van de FIFO buffer.

Zeer populair zou je kunnen stellen dat de Tx handshakesignalen een klepje besturen onder een trechter vol met bytes die staat boven het shift out register!

Algemene opmerking over de schakeling

Behalve de Tx clock, massa en de +5 V voeding zijn alle in het FIFO schema genoemde of vermelde IC pen nummers van de UART NIET meer doorverbonden met de modemschakeling. Ze worden slechts "lokaal" gebruikt op het hulprintje.

Let op: Als op pen 19 in "uw" modem een C-MOS ingangspoort zit i.p.v. een TTL poort, zoals in "Telektor", dan voor de netheid deze toch maar doorverbinden. Maar pen 18 zeker NIET.

Welke pennen zijn dan zeker wel doorverbonden van de 40 polige voet naar de 40 polige stekker:

1	Vcc +5V voeding	24	EOC end of char.
2	NC of -12 voeding	25	SO shift out (75Bd)
3	massa / GND	34	CS
4	RDE rec. data enable	35	NP
13	PE parity error	36	TSB
14	FE framing error	37	NB1
15	OR overrun error	38	NB2
16	SWE status word enable	39	EPS
17	RCP Rx clock puls	40	Tx clock
20	SI shift in (1200 Bd)		

HET (IN) BOUWEN

Om de print zo compact mogelijk te houden is een dubbelzijdige print ontworpen. Deze hoeft niet doorgemetaliseerd te worden, doorverbindingen van onder naar boven vinden ook niet plaats via de IC pennen, maar via extra draadjes (die dan eerst aan beide kanten worden omgebogen en platgedrukt, anders vallen ze eruit bij het solderen), of via de componenten. Er kunnen dus gewone IC voetjes gebruikt worden. Op de FIFO-2 print kunnen de voedingsontkoppel condensatoren tevens gebruikt worden als doorverbinding, je kunt ook aparte draadjes gebruiken hiervoor en de C's bovenop niet solderen, het mag ook alletwee. Dubbele zekerheid. De ontkoppel C die het dichtst bij de handshake IC's zit moet trouwens ALTIJD onder en boven gesoldeerd worden. Niet vergeten. De doorverbindingen en componenten moeten echter wel gesoldeerd worden voordat er voetjes op de print komen. Anders kun je er NIET meer bij komen.

Voordat de 40 polige flatcable (of DIP) stekker gemonteerd wordt moeten alle kontaktlipjes die niet gesoldeerd worden op het printje afgeknipt en weggeveijld worden. Dit om te voorkomen dat er alsnog ergens contact gemaakt wordt. Bij gebruik van een C-MOS ingangspoort op de modem print, zou pen 19 toch doorverbonden kunnen worden. Anders zweeft deze ingang en is dan ongedefinieerd, ergens mee verbonden vindt C-MOS prettiger. In "Telektor" is dit niet nodig want daar is een TTL ingang gebruikt. Omdat er in het midden van de 40 polige stekker zoveel kontakten niet gesoldeerd worden, is het zelfs mogelijk een 24 polige stekker te gebruiken die ergens in het midden doorgesneden wordt met een surgical mesje aan de soldeerbout. Leg de pleisters en brandzalf vast klaar. Deze twee stukjes prikken we, na schoonvrijen, in een 40 polig voetje met gedraaide kontakten voor de juiste maat bij het in positie solderen. De "unit" zit dan later wel iets minder vast! Gebruik voor het vast solderen een soldeerbout met een vreselijk klein puntje, want je bakt alles zo aan elkaar. Meet alle pennen na op sluiting! En leg desoldeermateriaal ook maar klaar voor het verwijderen van overdadig tin. Verder zijn er niet al te veel problemen te verwachten.

Als alles klaar is.....

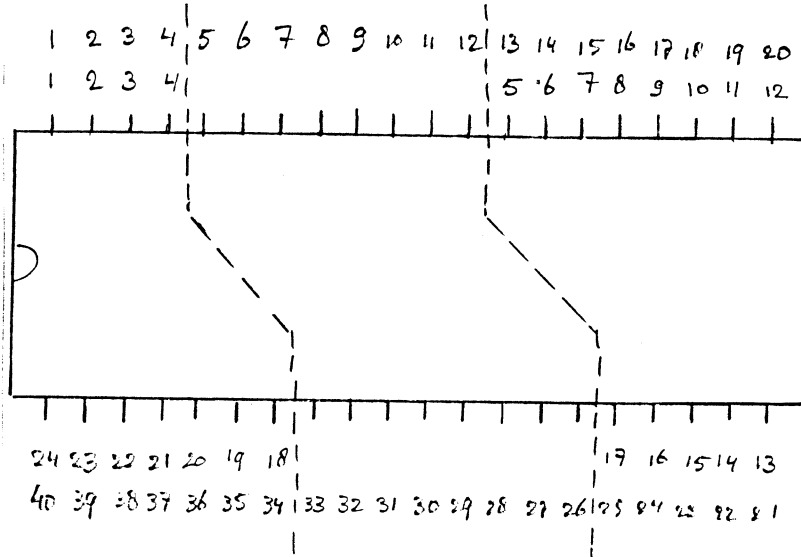
De UART gaat uit de modemprint en in de FIFO print. In de lege voet in het modem moet nu de FIFO print met UART. In vrijwel alle gevallen zal het printje, indien het zo in de voet gestoken wordt, andere componenten aan de onderkant raken. Dit lossen we op door een extra 40 pens voetje (dezelfde van het soldeer malletje!) te gebruiken als verhoging. Nogmaals: kies hiervoor niet een te goedkoop voetje want dit zal slecht contact maken in de modem print omdat de pennetjes te dun zijn. Zo'n professioneel voetje met gedraaide bus kontakten is uitstekend. Helaas rekt dit de bussen van het onderliggende voetje weer uit, je zult hem meteen voorgoed moeten laten zitten.

En dan modemen maar, succes,

W.A.J. Geeraert (Walter)
Frans Halslaan 2A
4382 RG Vlissingen

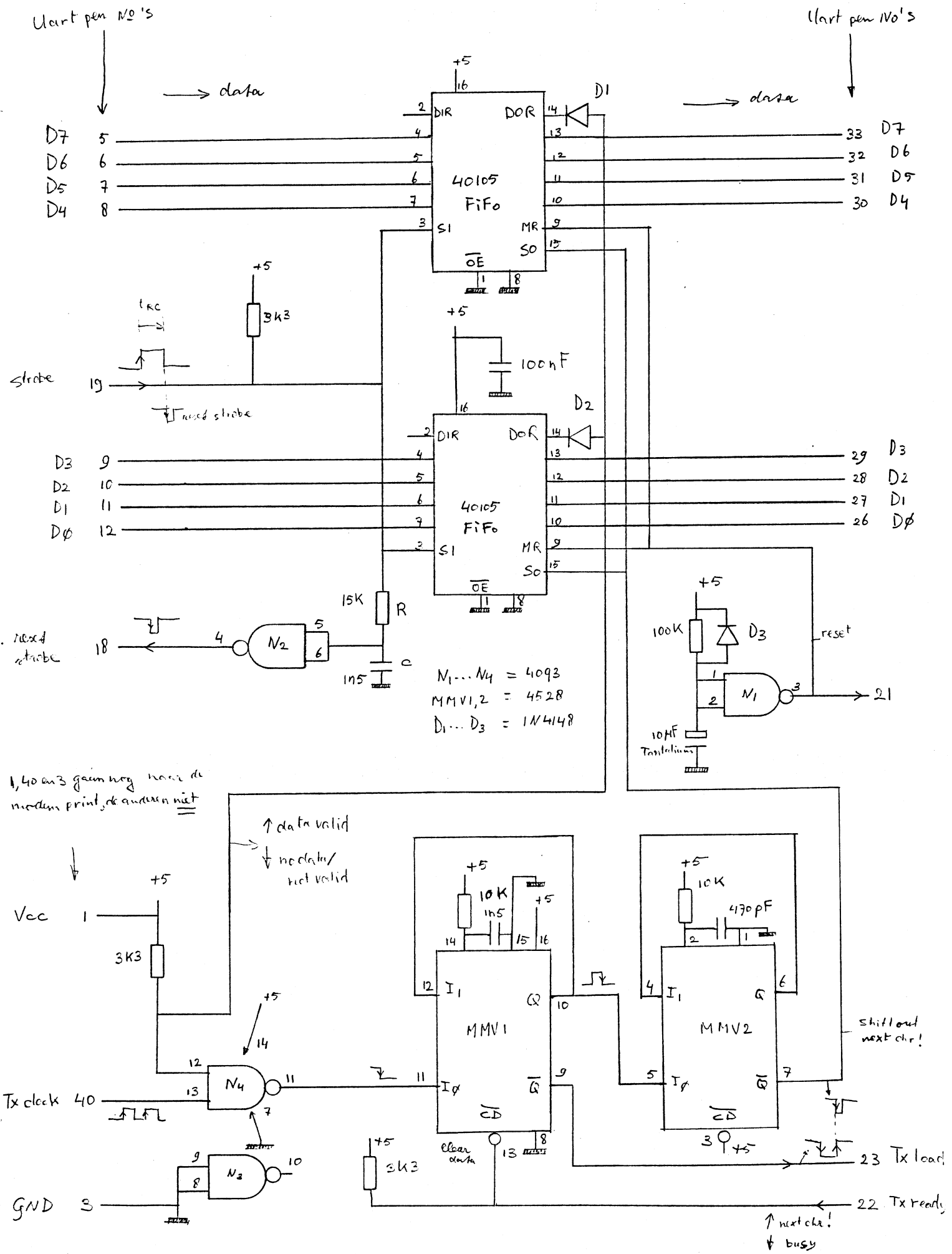
Tel 's avonds: 01184-19768 (met mate a.u.b.)

Gebruik van
twee stukjes
24 - polige
flatcable
stekker

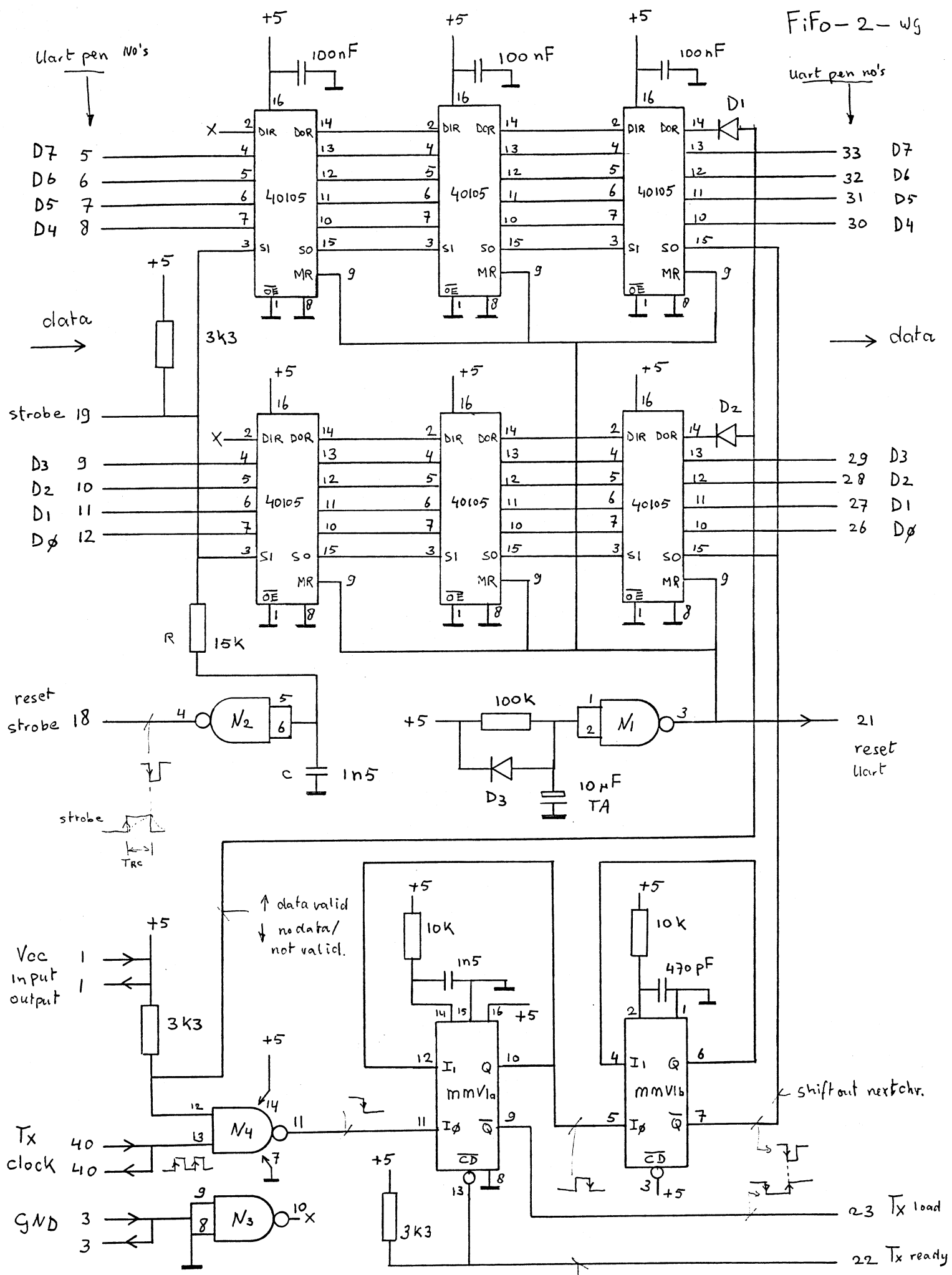


Alle hier genoemde kaartpen no's, behalve 1,3 en 40, gaan alleen naar de kaart en niet naar maar de modem print. De overigen wel van stekker naar busunit!

Fifo-1



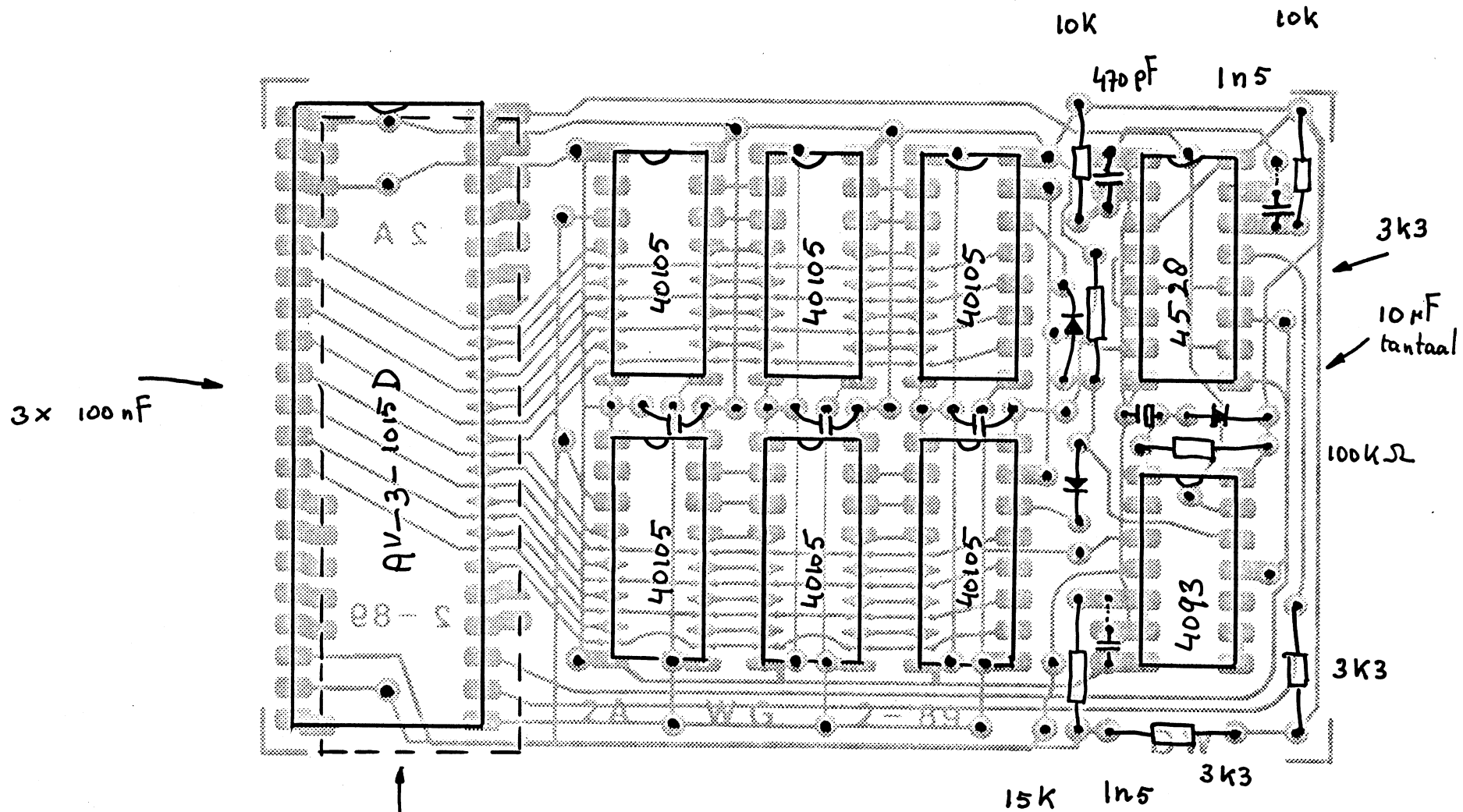
Fifo-2-wg



Alle genoemde Uart pen no's, behalve 1, 3, 4, gaan alleen naar de Uart!! . De overigen wel van steker naar bus-waait

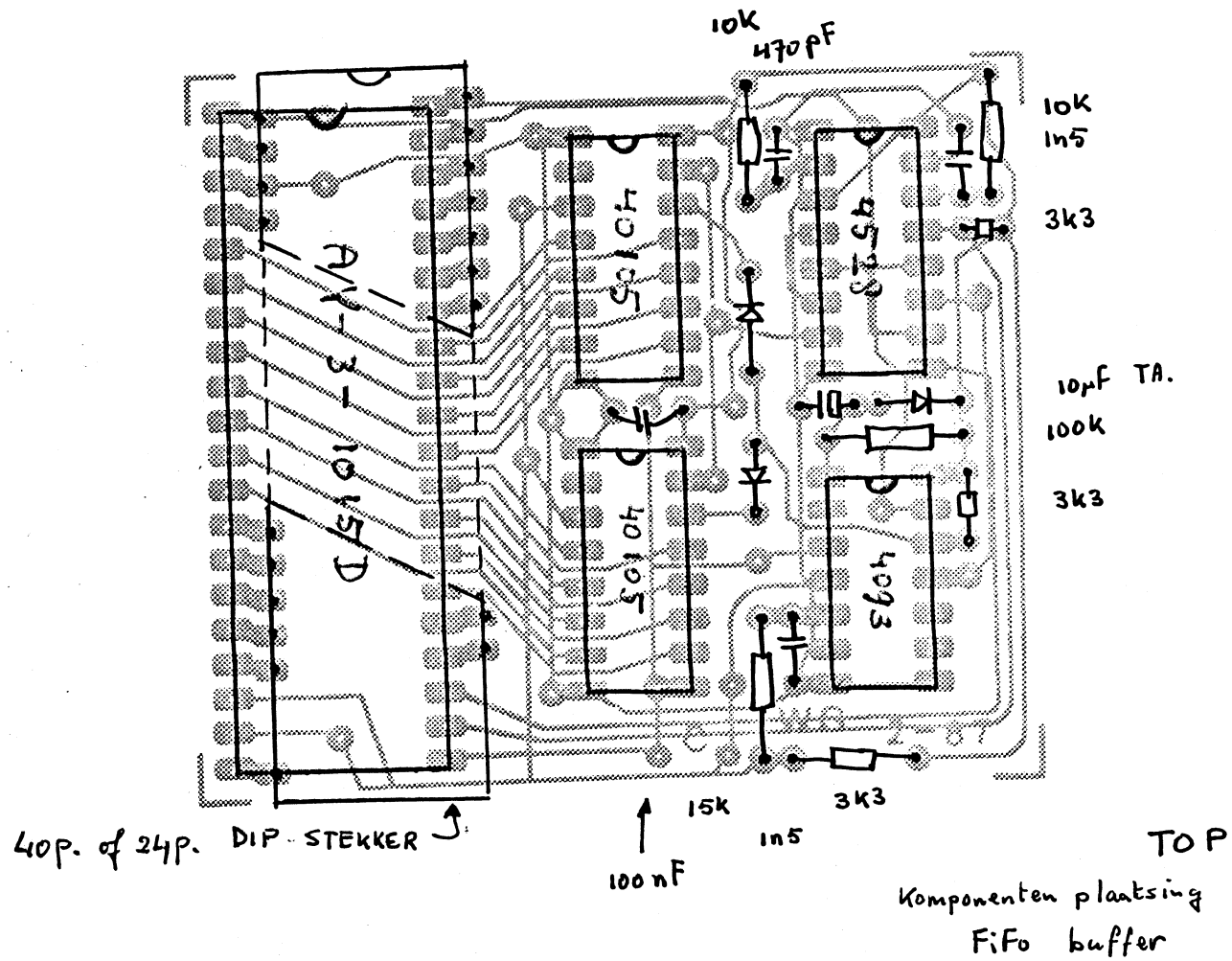
↑ next chr!
 ↓ busy
 D1...D3 = 1N4148
 N1...N4 = 4093
 mmV1a,1b = 4528

Fifo - 2

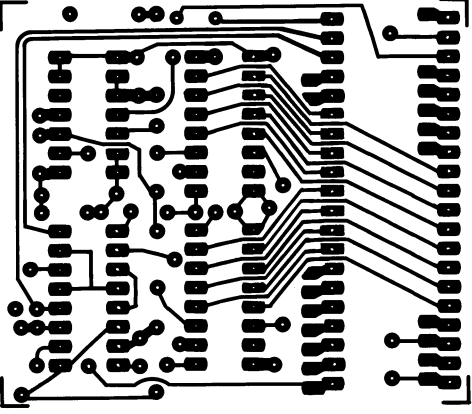


40 pins DIP stekker
(of flat cable stekker)

Onderdelen plaatsing en
doorverbindingen onder/boven kant
EERST dit solderen, dan de IC voetjes!

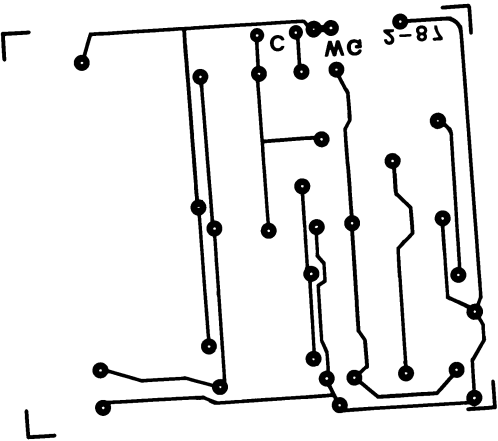


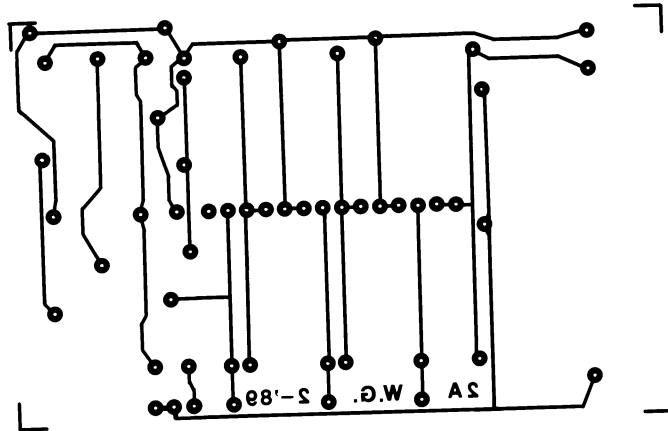
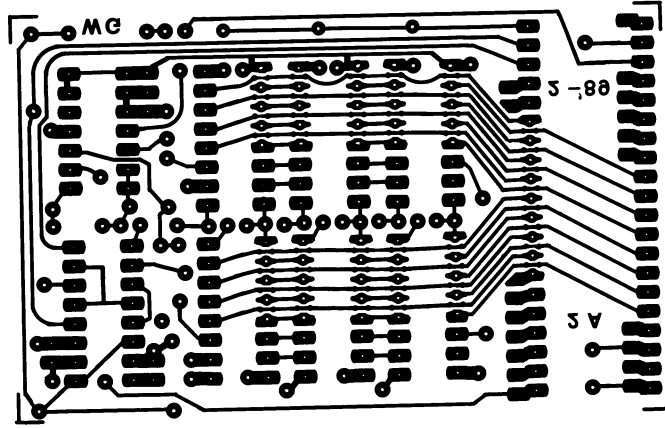
9
7
8
9
0
1
2
3
4
5
6
7
8
9



HUNSON
ALPHABET

7
8
9
10
11
12
13
14
15
16
17
18
19
20





17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

