

monitor expansion

EXPAN vers. 1.01

>
>
> COPYRIGHT © 1981
> BY
> GBR. VAN MONTFORT
> -

MONITOR EXPANSION PAC VERSION 1.01

Made by Gebr. van Montfort
 COPYRIGHT (C) 1981

THE EXPAN COMMANDS ARE:

BA CL CO CR DA DB DC DF DM DR DS DU
 EN ER EX FI FL GO HE IN LI LO MO OF
 OT OV PP PR RB SA SB SD SE SL SW TE

DU	dump	DU	XXXX	(XXXX)
DA	dump ASCII	DA(G)	XXXX	(XXXX)
DM	dump mnemonics	DM	XXXX	(XXXX)
EN	enter	EN	XXXX	
SD	search data	SD	(XXXX)	(XXXX)
MO	move	MO	XXXX	XXXX (S) XXXX
FL	fill	FL	XXXX	XXXX data
SB	set breakpoint(s)	SB	.	XXXX(XXXX(XXXX(XXXX(XXXX))))	
RB	reset breakpoints	RB		(XXXX)
DB	dump breakpoints	DB		
CO	continue	CO		(XXXX)
DR	dump registers	DR		
ER	enter registers	ER	reg=XX	(reg=XX ...)
DS	dump stack	DS		(n)
EX	exquite	EX(Q)	XXXX	XXXX
GO	go	GO		XXXX
LO	load	LO(G)	(NAMEX)	(# (XXXX))
FI	files	FI		(#)
DF	dump fileheader	DF		
SA	save file	SA	NAMEX XXXX (S)XXXX	(#)
SAF	save previous file	SAF	(NAMEX)	(# (XXXX))
CR	create batch # 1	CR		
BA	batch # 1	BA		
LI	list BA commands	LI		
OV	batch over	OV		
SE	set command	SE		
PR	prompt PR=?(CR)	PR		
TE	test	TE	XXXX	YYYY (C)
HE	hex (+) (-) (%=D)	HE	XXXX	((+) (-) (%) (XXXX))
SW	switch half screen	SW		
CL	clear left screen	CL		
OF	expan off	OF		
PP	prom pac	PP		(X)
IN	input	IN		(XX)
OT	output	OT		(XX) YY
DC	dump commands	DC		

COPYRIGHT Gebr. van Montfort,
 Smedestraat 13,
 6411 CR HEERLEN.

Inleiding:

Het programma "EXPAN" is geschikt voor het onderzoeken van machinetaal programma's. Er zitten een aantal mogelijkheden in, om met de geheugeninhoud te manipuleren en om bepaalde byte's te zoeken.

Bij de nu volgende beschrijving zijn ook de standaard monitor commando's mee omschreven. De monitor commando's EN SA en MO zijn echter iets veranderd.

De commando's staan in de linker kolom; alleen de eerste twee letters zijn noodzakelijk. De parameters van elk commando staan in de rechter kolom; parameters tussen haakjes zijn voor extra functies.

DU dump DU XXXX (XXXX)

Geeft de geheugeninhoud hexadecimaal weer. Indien U een adres heeft opgegeven, alleen de geheugeninhoud van dat adres, als U twee adressen heeft opgegeven, dan krijgt U alle inhouden van het eerste tot het tweede adres.

DA dump ASCII DA(G)..... XXXX (XXXX)

Geeft de inhoud in ASCII-code weer. Indien U een adres heeft opgegeven, van dit adres totdat U onderbreekt. Heeft U twee adressen opgegeven, dan tot het tweede adres. Gebruikt U de letter G in het commando, dan worden de grafische-codes omgezet in ASCII. Niet printbare ASCII code's worden weergegeven met een punt.

DM dump mnemonics DM XXXX (XXXX)

Geeft de geheugeninhoud in mnemonics weer. Indien U een adres heeft opgegeven, vanaf dit adres totdat U onderbreekt. Heeft U twee adressen opgegeven, dan tot het tweede adres. Breakpoints (zie verder) worden met "--" weergegeven.

EN enter EN XXXX

Met dit commando kunt U geheugen inhouden onderzoeken en veranderen. Als U RETURN alleen intoetst krijgt U het volgende adres, met ^ het vorige adres. U kunt hexadecimaal code invoeren, maar ook ASCII code door de letter tussen quotes ("") te typen. Als U vooruit wil springen typt U desbetreffende aantal keren X op een regel in. Indien terug gebruik dan een aantal keren ^ tekens. Relatieve jumps worden door het programma zelf uitgerekend, indien U een letter R met het desbetreffende adres intypt. Onderstaand een aantal voorbeelden

EN FC00 (CR)	Dit is het commando voor het invoeren van data vanaf adres FC00.
FC00 40:	De "Sorcerer" print dan het adres gevolgd door de inhoud hiervan.
FC01 40:00 (CR)	U heeft nu 00 ingevoerd op adres FC00, en U bent nu op het volgende adres aangekomen. Stel U wilt weer terug naar het vorige adres, type dan: ^ (CR).
FC01 40:	
FC01 40: ^ (CR)	
FC00 00: (CR)	U kunt verder naar het volgende adres door return in te toetsen.
FC01 40:/ (CR)	Slash (/) onderbreekt het "enter" commando.
EN 0000 (CR)	Voor het invoeren van data op adres 0000
0000 00:"ABCDE" (CR)	U heeft nu de ASCII waarde van A ingevoerd op adres 0000, ASCII B op adres 0001 etc. De sorcerer komt nu terug met adres 0005
0005 00:^ ^ ^ ^ ^ (CR)	Door 5 maal ^ te gebruiken bent U weer bij adres 0000
0000 41:(CR)	
0000 42:(CR)	
0000 43:(CR)	
0000 44:(CR)	
0000 45:/ (CR)	
EN 0010	Stel U heeft een machinetaal programma op pagina 0 en U wilt op adres 0010 en 0011 een relatieve jump naar adres 0040 invoeren.
0010 00:18 R0040 / (CR)	

SD search data SD(XXXX) (XXXX)

Met dit commando kunt U in het geheugen zoeken naar bepaalde byte's. Indien U geen adres opgeeft zoekt de "sorcerer" het gehele geheugen hierna af. Indien U een adres opgeeft, vanaf dit adres tot FFFF en als U twee adressen opgeeft alleen tussen deze twee adressen. U kunt de data die U zoekt hexadecimaal of met ASCII opgeven. Als er in de reeks data een byte is dat onbekend is kunt U dit byte aangeven met X. Onderstaand enige voorbeelden hiervan.

SD E000 EFFF "ERROR" Met dit commando vindt U alle adressen in de monitor-rom waar ERROR staat.

SD "ERROR" Nu vindt U alle adressen waar ERROR staat.

SD CD A2 E1 Dit zoekt alle adressen waar achtereenvolgens CD A2 E1 staat.

SD CD X E1 Op deze manier zoekt de sorcerer alle adressen waar CD staat en tevens twee adressen verder E1 staat.

MO move MO XXXX XXXX (S) XXXX

Met dit commando kunnen blokken data verschoven worden, zowel naar hogere als naar lagere adressen. Deze adressen mogen ook overlappend zijn, dit in tegenstelling tot het monitor commando MO. Er moeten drie parameters meegegeven worden, die op twee manieren ingevoerd kunnen worden. De eerste parameter is altijd het beginadres waar het blok vandaan komt. De tweede en derde parameter is bij de S optie respectievelijk het adres waar het naar toe verschoven moet worden en de derde het aantal byte's. Zonder S is de tweede parameter het einde van het eerste blok en de tweede het adres waar naar toe. Voorbeelden:

MO 0000 0020 0010 Verplaatst de bytes van adres 0000 tot en met 0020 naar adres 0010 tot en met 0030

MO 0010 0000 S0021 Verplaatst de bytes van voorbeeld weer terug.

FL fill FL XXXX XXXX data

Om geheugen adressen met bepaalde bytes te vullen kan dit commando gebruikt worden. Vanaf het eerste adres tot en met het tweede adres wordt gevuld met "data". Deze "data" kan een hexadecimale- of ASCII waarde hebben. Onderstaand voorbeelden:

FL 0000 0080 00 Vult de adressen 0000 tot en met 0080 met 00

FL F080 F0FF "A" Vult de adressen F080 tot en met F0FF met de ASCII waarde van "A"

FL F080 F0FF "HALLO" OD Vult de adressen F080 tot en met F0FF met "HALLO" en OD.

BREAKPOINTS

Het programma " EXPAN " heeft enige mogelijkheden met het debuggen (fouten opsporen) van machinetaal programma's. Het kan maximaal 5 breakpoints bijhouden en kan een programma opstarten of voortzetten, waarbij de inhouden van de registers opgegeven kunnen worden.

Attentie: bij gebruik van breakpoints worden de adressen 0038, 0039 en 003A gebruikt en bovendien de adressen F050 t/m.F077.

SB set breakpoint(s) SB . XXXX(XXXX(XXXX(XXXX(XXXX))))

Hiermee kunnen breakpoints (minimaal 1, maximaal 5) gezet worden. Dit moet wel een adres zijn waar een opcode staat, dus als b.v. op adres 1020 staat: 3E 80, wat betekend LD A,80 dan mag het breakpoint niet op adres 1021 gezet worden. Met het commando DM kunt U kijken waar precies een breakpoint mag staan.

RB reset breakpoints RB (XXXX)

Met het commando RB kunt U alle breakpoints verwijderen. Zonder parameters worden alle breakpoints verwijderd, met een of meer parameters worden echter alleen deze specifiek gereset.

DB dump breakpoints DB

Toont alle breakpoints

CO continue CO (XXXX)

Met dit commando kunt U na een breakpoint weer verder gaan. Het breakpoint waar gebroken is, wordt automatisch gereset. Indien een parameter extra wordt ingevoerd, wordt op dat adres een nieuw breakpoint geplaatst.

DR dump registers DR

Dit geeft de registerinhouden weer na/voor het uitvoeren van een breakpoint. Op de bovenste regel staan de primaire- en op de tweede regel de secundaire registers.

ER enter registers ER reg=XX (reg=XX ...)

De register-inhouden kunnen met dit commando veranderd worden. Er kunnen ook register-paren worden ingevuld. De registers zijn:

A, F, B, C, D, E, H, L, A', F', B', C', D', E', H', L' en de 2-bytes registers: BC, DE, HL, BC', DE', HL', IX, IY, SP en PC

ER A=12 Register A wordt 12

ER A=23 BC=1FFF

ER HL=2000 HL'=1000 F=3A

DS dump stack DS(n)

Met het commando DS worden bytes, die op de stack staan in groepen van twee getoond. Men kan eventueel het aantal (n) groepen bytes meegeven, maximaal 10

EX exiquite EX(Q)..... XXXX XXXX

Met dit commando kunt U een machinetaal programma opstarten. Als U twee parameters opgeeft, is het eerste het startadres en het tweede de stack-pointer. Als U een parameter opgeeft is de eerste het startadres, de stackpointer blijft dezelfde als bij DR Het returnadres wordt echter wel op de stack gezet. Gebruikt U het commando EXQ, dan wordt ook de programcounter(PC) ingevuld. OPGELET, het return-adres wordt niet op de stack gezet, het programma dat U aanroept moet dan ook op het einde naar de monitor jumpen, dus beslist niet met de instructie RET(return).

De Q betekent in dit commando QUIT voor het onderdrukken van errormeldingen. U moet hierbij zeer goed opletten om een crash te voorkomen.

GO go GO XXXX

Met GO kunt U een machinetaal routine aanroepen. Op het einde van deze routine moet of de instructie RET(return) staan of JP E003(jump monitor warm start).

LO load LO(G).....(NAMEX) (# (XXXX))

Hiermee leest de sorcerer het programma genaamd NAMEX in het geheugen, van recorder aangegeven door het recodernummer # (indien niet opgegeven van recorder een). Het programma wordt ingelezen-indien niet opgegeven- op het adres in de fileheader. Als er geen programma naam meegegeven is wordt het eerstvolgende programma ingelezen. Met toevoeging van de letter "G" na "LO" wordt het programma direct na het inlezen opgestart, met als startadres het GOADDR in de fileheader. B.v:

LO TEST1 Leest het programma TEST1 van recorder # 1, beginnend op het adres in de fileheader.

LO TEST1 2 Leest het programma TEST1 van recorder # 2, beginnend op het adres in de fileheader.

LO TEST1 1 0100 Leest het programma TEST1 van recorder # 1, beginnend op het adres 0100

LOG Leest het programma TEST1 van recorder # 1, beginnend op het adres in de fileheader.

FI files FI(#)

Leest alle fileheaders van de opgegeven recorder, indien niet opgegeven van recorder # 1.

DF dump fileheader DF

Geeft de laatst ingelezen fileheader weer. Met het SET-commando (zie verder) kan de fileheader veranderd worden. Deze fileheader wordt gebruikt bij het SAF-commando (zie hieronder)

SA save file SANAMEX XXXX (S)XXXX (#)
SAF save previous file SAF(NAMEX) (# (XXXX))

Zonder "F" schrijft dit commando de geheugeninhoud op band, met de naam NAMEX, vanaf het eerste adres t/m het tweede adres, of vanaf het eerste adres met de bloklengte aangegeven door het adres achter de "S". Indien niet opgegeven van recorder # 1

Met "F" wordt die geheugen-inhoud weggeschreven volgens de laatst ingelezen fileheader(zie commando DF). Indien het laatst gelezen programma op een ander adres is ingelezen, dan in de fileheader staat, kan door dit adres weer toe te voegen het programma weer correct worden weggeschreven. Zie voorbeelden.

SA TEST1 1000 10FF 2	Schrijft het programma vanaf adres 1000 tot en met 10FF op band met de naam TEST1 naar recorder # 2
SAF	Schrijft het laatst ingelezen programma met dezelfde fileheader beginadres en lengte op band naar recorder # 1 (dit programma wordt dus gecopieerd).
SAF TEST2 2	Schrijft het laatst ingelezen programma op band met de naam TEST2 en voor de rest dezelfde fileheader, naar recorder # 2
SAF 1 0000	Stel U heeft het programma TEST3 ingelezen met het commando: LO TEST3 2 0000 Hiermee wordt het programma TEST3 gecopieerd naar recorder # 1 ongeacht het startadres van dit programma.
SA TEST1 1000 S100	Hiermee wordt het programma TEST1 gecopieerd vanaf adres 1000 met een bloklengte van 100
SE set command	SE
SE S=XX	Veranderd de print vertraging met XX, 00 is de laagste en FF is de hoogste vertraging.
SE T=X baudrate	0=1200;1=300 (2=1200;3=300 RS232)
SE F=XX	Vult file-type in fileheader. De verandering is waarneembaar met het commando DF. Programma's waarvan file-types groter zijn dan 80, worden niet automatisch opgestart bij het commando LOG.
SE X=XXXX	Plaatst automatisch-opstartadres in fileheader. De verandering is waarneembaar met het "DF" commando.
SE O=Outputvektor naar....	
SE O=V	video
SE O=P	parallel output poort
SE O=L	centronics printer
SE O=S	tape output
SE O=XXXX	adres XXXX
SE I=Inputvektor naar....	
SE I=K	keyboard
SE I=P	parallel input poort
SE I=S	tape input
SE I=XXXX	adres XXXX

CR create batch # 1 CR
Maakt een batch-band aan op recorder # 1

BA batch # 1 BA
Hierdoor worden batch-commando's van recorder # 1 ingelezen en uitgevoerd. Uitsluitend oorspronkelijke monitor-commando's!

LI list BA commands LI
Hierdoor worden batch-commando's gelist.

OV batch over OV
Dit moet als laatste commando op de batch-band staan. Het beëindigt inlezen en uitvoeren van batch-commando's.

PR prompt=?(CR) PR
Veranderd monitor prompt in elk gewenst teken.

TE test TE XXXX YYYY (C)
Test elk bit van de ram, vanaf het eerste adres tot het tweede adres. Tijdens het testen knippert "*". Indien geen melding dan is het geheugen in orde; anders wordt BAD of OK gemeld voor elk bit. Met "C", gaat de test continu door, na elke doorloop komt de melding PASS-COMPLETED.

HE hex (+) (-) (%=D) HE XXXX ((+) (-) (%) (XXXX))
Hexadecimaal rekenen zowel optellen als aftrekken en nogwel achtereenvolgens is mogelijk. Ook decimale getallen kunnen, met "%" als voorvoegsel, bij deze twee rekenkundige bewerkingen worden gebruikt. De uitkomst komt in drie-voud, hexadecimaal, decimaal en twee complement.

HE 1000 + 9000 (CR) antwoord: A000
HE %255 + FF00 (CR) antwoord: FFFF
HE 0000 - 1 (CR) antwoord: FFFF 66563 -1

SW switch half screen SW
Dit commando maakt het mogelijk om de linker beeldhelft, die meestal het belangrijkste is, naar de rechterkant te verplaatsen. Een schoon beeldhelft blijft links achter. In de RUN/STOP mode heeft het intoetsen van de letter "S" dezelfde uitwerking.

CL clear half screen CL
Alleen linker beeldhelft schoonmaken, is vaak ook nuttig. In de RUN/STOP mode heeft het indrukken van de clear toets dezelfde uitwerking.

OF expan off OF

Om terug te keren in de monitor

PP prom pac PP (X)

Start ROM PAC, indien geen parameter, dan warm start, anders koude start.

IN input IN (XX)

Leest inputpoort XX. Indien geen parameter opgegeven is, wordt inputpoort FF gezet.

OT output OT (XX) YY

Zet YY op outputpoort XX. Bij een parameter wordt YY op outputpoort FF gezet.

DC dump commands DC

Toont alle actuele commando's

E R R O R meldingen:

Invalid parameter

command

breakpoint onbekend

tape crc error

no ram in rom

no file geen file aanwezig

no room meer dan 5 breakpoints

jr error buiten bereik van relatieve jump

stack geen continue mogelijk

Opmerkingen:

Zowel hoofd- als kleine letters kunnen worden toegepast.

Speciale mogelijkheden met gebruik van de dubbele punt (:)

b.v.: DA 1000 1200:L schakelt bij uitvoering van dit commando de centronics printer aan en weer uit. Ook met andere commandos mogelijk.

Leeg tussenvel

Geachte sorcerer bezitter,

U bent hiermee in het bezit gekomen van een zeer geavanceerd monitor programma, waarbij de mogelijkheden voor directe toepassing legio zijn. Het behoeft geen betoog dat dit programma met zeer veel inspanning en vooral tijd door ons is ontworpen. Wij verzoeken U dan ook begrip op te brengen voor de kleine bijdrage ad f 25,00, ter ondersteuning van ons werk voor U. Mocht U dit programma op andere wijze dan via ons ontvangen hebben, dan verzoeken wij U beleefd het bedrag alsnog aan ons te doen toekomen, s.v.p. met vermelding: bijdrage expan middels een cheque/girobetaalkaart etc.

Mocht U echter een uitgewerkt idee voor uitbreiding van dit programma hebben, dan verzoeken wij U dat aan ons toe te zenden. Indien dit idee voor algemene toepassing bruikbaar mocht zijn, dan ontvangt U, het door U betaalde bedrag weer retour.

In elk geval verzoeken wij U, onderstaande strook aan ons op te sturen, zodat wij U op de hoogte kunnen houden van uitbreidingen wijzigingen of andere ontwikkelingen op sorcerer gebied.

met vriendelijke groeten,

giro : 3397863

Gebr. van Montfort,
Smedestraat 13,
6411 CR HEERLEN.

H. van Montfort
Smedestraat 13,
6411 CR HEERLEN.

.....
Expan 1.01

.....
datum

Naam:

Adres:

Postcode: Plaats:

Telefoon:

Opmerkingen:

Leeg tussenvel

L.S.

Met genoegen ontvingen wij het door U teruggezonden briefje van de "EXPAN" handleiding.

Uit enige reacties is gebleken dat het gebruik van "breakpoints" als moeilijk ervaren wordt. Daarom is in dit supplement een voorbeeld uitgewerkt.

Bovendien kan "EXPAN" zelfs in EPROM gezet worden. De RELOCATOR (het eerste stuk van EXPAN) zal EXPAN van standaard adres 1000 t/m 2000 hex, verplaatsen naar de gewenste adressen en opnieuw opstarten. Om EXPAN geschikt te maken voor EPROM moet men de volgende instructies uitvoeren:

Zet de computer uit/ verwijder het ROM PAC/ zet de computer weer aan.

EN 38 (CR)

0038: C3 03 E0 / (CR)

(Bij het opstarten van EXPAN zal naar een lokatie in het pack gesprongen worden. Als er geen pac aanwezig is, ziet de microprocessor de instructie FF hetgeen betekend CALL 0038H. Vanaf hier wordt naar de monitor gesprongen.)

Daarna met LOG de EXPAN RELOCATOR van cassette inlezen. (zonder ROM PAC !!)

Zodra de RELOCATOR geladen en opgestart is, het adres opgegeven (bv. D000) gevolgt door (CR).

Hierna komt men terug in de monitor en staat het programma gereed op 1000 t/m 1FFF voor de eprom-programmer.

Verder wensen wij U veel plezier met ons programma.

Namens de gebroeders van Montfort.

H.L.L. van Montfort.

1 bijlagen.

Gebr. van Montfort
Smedestraat 13, 6411 CR Heerlen, tel. 045-712604, giro 3611054

Interupts

Het programma "EXPAN" heeft op elk moment een geldige stack (SP), daarom mag op elk moment een interupt gegeven worden.

Bij interupts en breakpoints zijn er twee moeilijkheden:

- EXPAN maakt gebruik van adres 38H. 38H kan dus niet meer voor interupts gebruikt worden. De enigste manier om toch de breakpoints en interupts te gebruiken is de Z80 micro-processor in mode 2 te programmeren (zie Z80-handboek).
- Tijdens het uitvoeren van een "break" mag de interupt routine niet meer dan 12 byte-paren van de stack gebruiken.

Voor het debuggen van interupt-routines heeft "EXPAN" de volgende mogelijkheden:

INT (CR)	toon interupt status & register
INT E (CR)	enable interupt
INT D (CR)	disable interupt
INT I=XX (CR)	set XX in het I register
INT M0 (CR)	set interupt mode 0
INT M1 (CR)	set interupt mode 1
INT M2 (CR)	set interupt mode 2

Een uitvoerigere beschrijving over interupt mogelijkheden van de Z80-CPU is in elk Z80-handboek wel te vinden.

Voorbeeld toepassingen breakpoints.

EN 0000 (CR)

0000 00: 21 80 F1 (CR)	
0003 00: E5 (CR)	
0004 00: 06 10 (CR)	
0006 00: 36 "A" (CR)	invoer ASCII van "A"
0008 00: 10 R0006 (CR)	
000A 00: E1 (CR)	(N.B. graphics is niet mogelijk)
000B 00: 06 10 (CR)	
000D 00: 36 "B" (CR)	
000E 00: 10 R000D (CR)	invoer relatieve sprongopdracht
0011 00: C9 / (CR)	

DM 0000 0011 (CR)

Hiermee controleert U op correcte invoer van het programma.

0001: 2180F1	LD HL, F180
0003: E5	PUSH HL
0004: 0610	LD B, 10
0006: 3641	LD (HL), 41
0008: 10FC	DJNZ 6-\$
000A: E1	POP HL
000B: 610	LD B, 10
000D: 3642	LD (HL), 42
000F: 10FC	DJNZ 0D-\$
0011: C9	RET

SB 0004 0006 (CR) Set breakpoints op adres 4 en 6.

DM 0000 0011 (CR) Controleer of de breakpoints op de juiste adressen staan.

```

0001: 2180F1 LD HL,F180
0003: E5 PUSH HL
0004: --10 LD B,10
0006: --41 LD (HL),41
0008: 10FC DJNZ 6-$
000A: E1 POP HL
000B: 610 LD B,10
000D: 3642 LD (HL),42
000F: 10FC DJNZ 0D-$
0011: C9 RET

```

GO 0000 (CR) Het programma stopt op 0004

break at 0004

A	SZ.A.PNC	BC	DE	HL	
OD	10111001	F180	
..	
(SP)	IX	IY	SP	PC	
....91	..8D	0004	

Omdat U het GO commando gebruikt heeft, is voortzetting van het programma niet (met CO) mogelijk.
De stack is nl. gelijk aan die van de monitor, en kan dus door de monitor overschreven worden.

Beter is:

EX 0000 0100 (CR) De tweede parameter wordt toegerekend aan de stackpointer. Hierdoor krijgt de routine een aparte stack die debuggen mogelijk maakt.

DS 1 (CR) Toon de waarde die op de stack gezet is (tot max 9).

CO (CR) Het programma wordt voort gezet, en breekt weer op 0006. Het breakpoint op 0004 is uitgewist.

CO 0008 (CR) Het programma wordt voortgezet vanaf het laatste breakpoint. Tevens wordt een breakpoint op 0008 ingevuld. (N.B. na CO kan maar een breakpoint opgegeven worden.)

Enige losse kreten.

- SD 0000 00FF "ABC" (CR) Zoek vanaf 0000 t/m 00FF naar ABC

- SD 0 FF "ABC" zoek het hele geheugen af naar 00, FF en ABC. Alleen bij SD-commando heeft 00XY en XY een verschillende betekenis.

