

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE EDL
 OBJECT MODULE PLACED IN :F1:EDL.OBJ
 INVOCATION LINE CONTRCLS: DEBUG

LOC	OBJ	LINE	SOURCE
		1 +1	STITLE ('EDL - EXTERNAL DATA LINK')
		2	NAME EDL
		3	
		4	;;; Intel Corporation Proprietary Information. This
		5	; listing is supplied under the terms of a license
		6	; agreement with Intel Corporation and may not be copied
		7	; nor disclosed except in accordance with the terms of
		8	; the agreement.
		9	
		10	DGROUP GROUP DATA,STACKS
		11	CGROUP GROUP CODE
----		12	DATA SEGMENT byte PUBLIC 'DATA'
		13	
		14 +1	\$NOLIST INCLUDE (:F1:KAOS.DCA)
		25 +1	\$NOLIST INCLUDE (:F1:DLL.DCA)
		31	
		32	; SEGMENT STRUCTURE
		33	
----		34	SEGMNT STRUC
0000		35	KAOSMSGHDR DD ?
0004		36	SEG_LENGTH DW ?
0006		37	DL_DEST DW 3 DUP (?)
000C		38	DL_SOURCE DW 3 DUP (?)
0012		39	DL_TYPE DW ?
0014		40	DL_DATA DB ?
----		41	SEGMNT ENDS
		42	
		43	; COMMAND TYPE DEFINITIONS
		44	
0001		45	CMD_CONNECT EQU 01H
0002		46	CMD_DISCONNECT EQU 02H
0003		47	CMD_ADDMCID EQU 03H
0004		48	CMD_DELMCID EQU 04H
0005		49	CMD_TRANSMIT EQU 05H
0006		50	CMD_SUPPLYBUF EQU 06H
0008		51	CMD_READ EQU 08H
0009		52	CMD_READC EQU 09H
		53	
		54	; OTHER CONSTANTS
		55	
003C		56	MIN_PAC_LEN EQU 60
05EA		57	MAX_PAC_LEN EQU 1514
		58	
		59	; REQUEST BLOCK STRUCTURES
		60	
----		61	HEADER STRUC
0000		62	MIPHDR DW 6 DUP (?)
000C		63	PROCID DW ?
000E		64	CMD DB ?
000F		65	RESULT DB ?

LOC	OBJ	LINE	SOURCE		
0010		66	RESPSOC	DW	?
----		67	HEADER	ENDS	
		68			
----		69	CONNECT	STRUC	
0000		70	JUNK1	DW	9 DUP (?)
0012		71	DLTYPE	DW	?
----		72	CONNECT	ENDS	
		73			
----		74	ADDMCID	STRUC	
0000		75	JUNK2	DW	9 DUP (?)
0012		76	MCID	DW	3 DUP (?)
----		77	ADDMCID	ENDS	
		78			
----		79	TRANS	STRUC	
0000		80	JUNK3	DW	9 DUP (?)
0012		81	LEN	DW	?
0014		82	EXTPTR	DD	?
0018		83	EXTLEN	DW	?
001A		84	XDATA	DW	?
----		85	TRANS	ENDS	
		86			
----		87	SUPPLY	STRUC	
0000		88	JUNK4	DW	9 DUP (?)
0012		89	JUNK5	DW	?
0014		90	RDATA	DW	?
----		91	SUPPLY	ENDS	
		92			
----		93	READ	STRUC	
0000		94	JUNK6	DW	9 DUP (?)
0012		95	OBJECT	DW	?
0014		96	RETURN	DW	3 DUP (?)
----		97	READ	ENDS	
		98			
		99			
0000	????????	100	REQPTR	DC	?
0004	(3	101	MCIDBUF	DW	3 DUP (?)
	????				
)				
000A	??	102	DROPFLG	DB	?
		103			
		104			
----		105	DATA	ENDS	
----		106	CCDE	SEGMENT	BYTE PUBLIC 'CODE'
		107		ASSUME	CS:CGROUP,DS:DGROUP
		108		EXTRN	CQMIPSEND:NEAR,CQMIPCONNECT:NEAR
		109		EXTRN	CQMIPGETADDRESS:NEAR
		110			
		111	+1	\$EJECT	

;SAVE FOR REQUEST POINTER

;BUFFER FOR MULTICAST ID

;0 = NO DROPPED PACKETS, 1 = DROPPED PACKETS

```

LOC  OBJ                LINE    SOURCE
                                112    ;;;      MAIN EDL SERVICE PROCESS
                                113
                                114
                                115    ;      END OF LOOP -- RETURN MESSAGE TO RESP SOC
                                116
0000  B001                117    RETONE: MOV    AL,1          ;RETURN A ONE -- FAILURE
0002  EB02                118          JMP    SHORT RETAL
0004  32C0                119    RETZER: XCR   AL,AL        ;RETURN A ZERO -- SUCCESS
0006  C41E000C            R      120    RETAL:  LES   BX,REQPTR    ;RETURN VALUE IN AL
000A  2688470F            121          MOV    ES:[BX].RESULT,AL
000E  26FF7710            122          PUSH  ES:[BX].RESPSOC
0012  06                  123          PUSH  ES
0013  53                  124          PUSH  BX
0014  E80000              E      125          CALL  CQMIPSEND
                                126
                                127
                                128    ;;;      START OF MAIN PROCESS
                                129
0017  B80B00              R      130    X:     MOV    AX,OFFSET DGROUP:MBX1
001A  50                  131          PUSH  AX
001B  E80000              E      132          CALL  CQRECEIVE          ;GET NEXT REQUEST
001E  891E0000            R      133          MOV    WORD PTR REQPTR,BX ;SAVE REQUEST POINTER
0022  8C060200            R      134          MOV    WORD PTR REQPTR+2,ES
                                135
0026  268A470E            136          MOV    AL,ES:[BX].CMD    ;WHAT COMMAND?
002A  3C05                137          CMP    AL,CMD_TRANSMIT
002C  755E                138          JNZ   X10              ;IF NOT TRANSMIT
                                139
                                140    ;      TRANSMIT COMMAND
                                141
002E  268B4712            142          MOV    AX,ES:[BX].LEN    ;CHECK LENGTH
0032  26034718            143          ADD    AX,ES:[BX].EXTLEN
0036  70C8                144          JC    RETONE          ;IF OVERFLOW
0038  3DEB05                145          CMP    AX,MAX_PAC_LEN+1
003B  73C3                146          JNC   RETONE          ;IF TOO BIG
003D  3D3C00                147          CMP    AX,MIN_PAC_LEN
0040  72BE                148          JC    RETONE          ;IF TOO SMALL
                                149
0042  50                  150          PUSH  AX              ;SAVE TOTAL LENGTH
0043  FF360000              E      151          PUSH  CQDLLTXFREEMBX   ;GET A TRANSMIT BUFFER
0047  E80000              E      152          CALL  CQRECEIVE
004A  58                  153          POP   AX              ;TOTAL LENGTH
004B  53                  154          PUSH  BX              ;SAVE BUFFER OFFSET FOR WHEN WE SEND IT
004C  894704                155          MOV    [BX].SEG_LENGTH,AX ;SET LENGTH
004F  8D7F06                156          LEA   DI,[BX].DL_DEST
0052  1E                  157          PUSH  DS              ;SAVE STANDARD DS
0053  C51E000C            R      158          LDS   BX,REQPTR
0057  8B4F12                159          MOV    CX,[BX].LEN
005A  8D771A                160          LEA   SI,[BX].XDATA    ;SET BEGINNING OF USER DATA
                                161
005D  FC                  162          CLD
005E  D1E9                163          SHR   CX,1            ;SET UP FOR WORD MOVE
0060  7301                164          JNC   X1              ;TEST FOR ODD BYTE
0062  A4                  165          MOVSB          ;MOVE ODD BYTE
0063  F3                  166    X1:   REP   MOVSW      ;MOVE WORDS

```

LOC	OBJ	LINE	SOURCE
0064	A5	167	
0065	8B4F18	168	MOV CX,[BX].EXTLEN ;SET UP FOR SECOND MOVE
0068	51	169	PUSH CX ;SAVE REGS FOR CALL TO GETADDRESS
0069	06	170	PUSH ES
006A	57	171	PUSH DI
006B	C57714	172	LDS SI,[BX].EXTPTR ;SET UP FOR CALL
006E	1E	173	PUSH DS
006F	56	174	PUSH SI
0070	06	175	PUSH ES ;PUT DGROUP (ALREADY IN ES) IN DS
0071	1F	176	PCP DS
0072	E80000	177	CALL CQMIPGETADDRESS
0075	8CC0	178	MOV AX,ES ;PUT RESULT IN RIGHT REGS
0077	8ED8	179	MOV DS,AX
0079	8BF3	180	MOV SI,BX
007B	5F	181	PCP DI ;GET OLD REGS BACK
007C	07	182	PCP ES
007D	59	183	POP CX
		184	
007E	D1E9	185	SHR CX,1 ;DO THE MOVE
0080	7301	186	JNC X2
0082	A4	187	MCVSB
0083	F3	188	X2: REP MOVSW
0084	A5	189	
0085	1F	190	POP DS ;RESTORE DS
0086	E80000	191	CALL CQDLLTXSEND ;SEND THE PACKET (BUF PTR ALREADY ON STACK)
0089	E978FF	192	JMP RETZER ;DONE
		193	
008C	3C06	194	X10: CMP AL,CMD_SUPPLYBUF ;CHECK IF SUPPLYBUF COMMAND
008E	750C	195	JNZ SHCRT X20 ;IF NOT
		196	
		197	; SUPPLYBUF COMMAND
		198	
0090	B81B00	199	MOV AX,OFFSET DGROUP:MBX2 ;SEND TO MAILBOX
0093	50	200	PUSH AX
0094	06	201	PUSH ES
0095	53	202	PUSH BX
0096	E80000	203	CALL CQSEND
0099	E97BFF	204	JMP X ;LOOP
		205	
009C	3C01	206	X20: CMP AL,CMD_CONNECT
009E	7511	207	JNZ X30
		208	
		209	; CONNECT COMMAND
		210	
00A0	268B4712	211	MOV AX,ES:[BX].DLTYPE
00A4	86C4	212	XCHG AL,AH ;PUT BYTES IN THE RIGHT ORDER
00A6	50	213	PLSH AX
00A7	B82300	214	MOV AX,OFFSET DGROUP:MBX3
00AA	50	215	PUSH AX
00AB	E80000	216	CALL CQDLLCONNECT
00AE	E955FF	217	JMP RETAL ;RETURN WHATEVER VALUE CONNECT RETURNED
		218	
00B1	3C02	219	X30: CMP AL,CMD_DISCONNECT

LOC	OBJ	LINE	SOURCE
00B3	750D	220	JNZ X4C
		221	
		222	; DISCONNECT COMMAND
		223	
00B5	268B4712	224	MOV AX,ES:[BX].DLTYPE
00B9	86C4	225	XCHG AL,AH ;PUT BYTES IN THE RIGHT ORDER
00BB	50	226	PUSH AX
00BC	E80000	227	CALL CQDLLDISCONNECT
00BF	E942FF	228	JMP RETZER ;RETURN SUCCESS
		229	
00C2	3C03	230	X40: CMP AL,CMD_ADDMCID
00C4	7404	231	JZ X41
00C6	3C04	232	CMP AL,CMD_DELMCID
00C8	752C	233	JNZ X5C
		234	
		235	; ADD AND DELETE MCID COMMANDS
		236	
00CA	268B4712	237	X41: MOV AX,ES:[BX].MCID ;MOVE MCID TO LOCAL BUFFER
00CE	A30400	238	MOV MCIDBUF,AX
00D1	268B4714	239	MOV AX,ES:[BX+2].MCID
00D5	A30600	240	MOV MCIDBUF+2,AX
00D8	268B4716	241	MOV AX,ES:[BX+4].MCID
00DC	A30800	242	MOV MCIDBUF+4,AX
		243	
00DF	B80400	244	MOV AX,OFFSET DGROUP:MCIDBUF
00E2	50	245	PUSH AX
00E3	26807F0E03	246	CMP ES:[BX].CMD,CMD_ADDMCID
00E8	7506	247	JNZ X42
00EA	E80000	248	CALL CQDLLADDMCID
00ED	E916FF	249	JMP RETAL ;RETURN WHATEVER ADDMCID RETURNED
		250	
00F0	E80000	251	X42: CALL CQDLLDELMCID
00F3	E90EFF	252	JMP RETZER ;RETURN SUCCESS
		253	
00F6	3C08	254	X50: CMP AL,CMD_READ
00F8	7512	255	JNZ X6C
		256	
		257	; READ COMMAND
		258	
00FA	26FF7712	259	PUSH ES:[BX].OBJECT
00FE	33C0	260	XOR AX,AX
0100	50	261	PUSH AX
0101	8D4714	262	LEA AX,ES:[BX].RETURN
0104	06	263	PUSH ES
0105	50	264	PUSH AX
0106	E80000	265	CALL CQDLLREAD
0109	E9F8FE	266	JMP RETZER
		267	
010C	3C09	268	X60: CMP AL,CMD_READC
010E	7406	269	JZ X61
0110	B8FF00	270	MOV AX,OFFH ;RETURN "UNKNOWN COMMAND"
0113	E9F0FE	271	JMP RETAL
		272	
		273	; READ AND CLEAR COMMAND
		274	

LOC	OBJ	LINE	SOURCE
0116	26FF7712	275	X61: PUSH ES:[BX].OBJECT
011A	33C0	276	XCR XCR AX,AX
011C	50	277	PUSH AX
011D	8D4714	278	LEA AX,ES:[BX].RETURN
0120	06	279	PUSH ES
0121	50	280	PUSH AX
0122	E80000	281	CALL CQDLLREADC
0125	E9DCF8	282	JMP RETZER
		283	
		284 +1	\$ EJECT

LOC	OBJ		LINE	SOURCE
			285	;;; PACKET-RECEIVED SERVICE PROCESS
			286	
			287	
C128	B82B00	R	288	R: MCV AX,OFFSET DGROUP:MBX3 ;GET NEXT PACKET
012B	50		289	PUSH AX
C12C	E80000	E	290	CALL CQRECEIVE
C12F	53		291	PUSH BX ;SAVE PACKET POINTER
			292	
C130	B81B00	R	293	MCV AX,OFFSET DGROUP:MBX2 ;GET A USER BUFFER
C133	50		294	PUSH AX
C134	E80000	E	295	CALL CQCRECEIVE
C137	81FBFFFF		296	CMP BX,OFFFH ;CHECK IF THERE IS A BUFFER
C13B	750A		297	JNZ R1 ;IF THERE IS A USER BUFFER
013D	E80000	E	298	CALL CQDLLRXRETBUF ;DROP THE PACKET (PACKET OFFSET STILL ON STACK)
C140	C6060A0001	R	299	MCV DROPFLG,1 ;RECORD THAT A PACKET HAS BEEN DROPPED
C145	EBE1		300	JMP R ;LOOP
			301	
0147	32C0		302	R1: XOR AL,AL ;SET RESULT FIELD TO DROPFLG; CLEAR DROPFLG
0149	86060A00	R	303	XCHG AL,DROPFLG
014D	2688470F		304	MOV ES:[BX].RESULT,AL
C151	5E		305	POP SI ;POP PACKET POINTER
0152	268B4710		306	MOV AX,ES:[BX].RESPSOC
0156	50		307	PUSH AX ;AND RE-ARRANGE STACK FOR FUTURE CALLS
0157	06		308	PUSH ES
0158	53		309	PUSH BX
0159	56		310	PUSH SI
			311	
015A	8D7404		312	LEA SI,[SI].SEG_LENGTH
015D	8B0C		313	MCV CX,[SI]
015F	8D7F12		314	LEA DI,[BX].LEN
0162	FC		315	CLD
			316	
0163	D1E9		317	SHR CX,1
0165	7301		318	JNC R2
0167	A4		319	MCVSB
0168	F3		320	R2: REP MOVSW
0169	A5			
016A	A5		321	MOVSW ;NEED TO MOVE 2 MORE FOR LENGTH FIELD
			322	
016B	E80000	E	323	CALL CQDLLRXRETBUF ;RETURN THE PACKET BUFFER
016E	E80000	E	324	CALL CQMIPSEND ;SEND THE USER BUFFER BACK
C171	EBB5		325	JMP R ;LOOP FOREVER
			326	
			327	+1 \$ EJECT

LOC	OBJ	LINE	SOURCE
		328	
		329	PUBLIC EDLSTART
0173		330	EDLSTART:
0173	B88C01	331	MCV AX,OFFSET CGROUP:OBJLIST
0176	1E	332	PUSH DS
0177	50	333	PUSH AX
0178	E80000	334	CALL CQCREATELIST
		335	
017B	B80100	336	MOV AX,1 ;CONNECT TO MIP PORT 1
017E	50	337	PUSH AX
017F	B80B00	338	MCV AX,OFFSET DGROUP:MBX1
0182	50	339	PUSH AX
0183	E80000	340	CALL CQMIPCONNECT
0186	C6060A0C00	341	MCV DROPFLG,0 ;CLEAR DROPFLG
018B	C3	342	RET
		343	
		344	
		345	
		346 +1	\$NOLIST INCLUDE (:F1:LOGGEN.MAC)
		445	
		446	; OBJECTLIST (OBJLIST)
		447 +1	
----		448 +1	CODE SEGMENT PUBLIC 'CODE'
018C		449 +2	OBJLIST EQU \$
----		450 +1	CODE ENDS
		451 +1	
		452	
		453	; MAILBOX (MBX1)
		454 +1	
----		455 +1	CODE SEGMENT PUBLIC 'CODE'
018C	06	456 +1	DB 6
018D	0B00	457 +2	DW OFFSET DGROUP:MBX1
----		458 +1	CODE ENDS
----		459 +1	DATA SEGMENT PUBLIC 'DATA'
000B	(8	460 +2	MBX1 DW 8 DUP (?)
	????		
)		
----		461 +1	DATA ENDS
		462	
		463	; MAILBOX (MBX2)
		464 +1	
----		465 +1	CODE SEGMENT PUBLIC 'CODE'
018F	06	466 +1	DB 6
0190	1B00	467 +2	DW OFFSET DGROUP:MBX2
----		468 +1	CODE ENDS
----		469 +1	DATA SEGMENT PUBLIC 'DATA'
001B	(8	470 +2	MBX2 DW 8 DUP (?)
	????		
)		
----		471 +1	DATA ENDS
		472	
		473	; MAILBOX (MBX3)
		474 +1	
----		475 +1	CODE SEGMENT PUBLIC 'CODE'
0192	06	476 +1	DB 6

LOC	OBJ		LINE	SOURCE	
0193	2800	R	477 +2	DW	OFFSET DGROUP:MBX3
----			478 +1	CCODE	ENDS
----			479 +1	DATA	SEGMENT PUBLIC 'DATA'
002B	(8		480 +2	MBX3	DW 8 DUP (?)
	????				
----			481 +1	DATA	ENDS
			482		
			483	;	PROCESS (6,X, 84)
			484 +1		
----			485 +1	CODE	SEGMENT PUBLIC 'CODE'
0195	02		486 +1	DB	2
0196	3300	R	487 +2	DW	OFFSET DGROUP:PCB_X
0198	0600		488 +2	DW	6
019A	1700	R	489 +2	DW	OFFSET CGROUP:X
019C	5E00	R	490 +2	DW	OFFSET DGROUP:STK_X
----			491 +1	CODE	ENDS
----			492 +1	DATA	SEGMENT PUBLIC 'DATA'
003B	(7		493 +2	PCB_X	DW 7 DUP (?)
	????				
----			494 +1	DATA	ENDS
----			495 +1	STACKS	SEGMENT PUBLIC 'STACKS' ;CREATE STACK
0000	(47		496 +2	DW	2FH DUP (?)
	????				
)				
	005E		497 +2	STK_X	EQU \$
----			498 +1	STACKS	ENDS
			499		
			500	;	PROCESS (5,R, 84)
			501 +1		
----			502 +1	CODE	SEGMENT PUBLIC 'CODE'
019E	02		503 +1	DB	2
019F	4900	R	504 +2	DW	OFFSET DGROUP:PCB_R
01A1	0500		505 +2	DW	5
01A3	2801	R	506 +2	DW	OFFSET CGROUP:R
01A5	BC00	R	507 +2	DW	OFFSET DGROUP:STK_R
----			508 +1	CODE	ENDS
----			509 +1	DATA	SEGMENT PUBLIC 'DATA'
0049	(7		510 +2	PCB_R	DW 7 DUP (?)
	????				
)				
----			511 +1	DATA	ENDS
----			512 +1	STACKS	SEGMENT PUBLIC 'STACKS' ;CREATE STACK
005E	(47		513 +2	DW	2FH DUP (?)
	????				
)				
	00BC		514 +2	STK_R	EQU \$
----			515 +1	STACKS	ENDS
			516		
			517	;	ENDLIST
			518 +1		
----			519 +1	CODE	SEGMENT PUBLIC 'CODE'
01A7	00		520 +1	DB	0
----			521 +1	CODE	ENDS

LOC	OBJ	LINE	SOURCE
		522	+1
		523	
		524	
		525	
		526	
----		527	CODE ENDS
		528	
		529	END

ASSEMBLY COMPLETE, NO ERRORS FOUND