MICRO.DOC

FOONLY F2 and F3
MICRO-PROGRAMMER'S MINI-MANUAL

|L

This file is MICRO.DOC --- description of F2 and F3 microcode and machine architecture, from programmer's point of view.

Section 1. Description of microcode on a field by field basis

CYLEN field   bits 00-03
This field specifies the cycle length of the microinstruction.

Assembler Mnemonic	Value	Meaning
TOO-SHORT	17	300 nsec
SHORT	16	350
NORM	15	400
C450	14	450
C500	13	500
C550	12	550
C600	11	600
C650	10	650
C700	7	700
	6	750
	5	800
	4	850
	3	900
	2	950
	1	1.00 microsec
LONG	0	1.05 microsec

Note: there other mnemonics for CYLEN field values...but they tend to change for different versions of the microcode. The best bet is to look at the microcode definition file CFDEF.SLO.

|L

COND field bits 04-08

Jump Concition Code. Depending on the contents of the J-CODE field, the truth value of the condition specified in the COND field may alter the actions of the microsequencer. Bit 04 (the high order bit of the field) is designated REV. It inverts the truth value of the condition. For every condition designated by X, there is a corresponding reversed condition designated by -X. The value of the reversed condition field is the non-inverted value plus 20 (octal).

Mnemonic	Value	Meaning
TRUE	0	the non-conditional condition
INTRPT	1	there is an interrupt request
MA-AC	2	the address in the MA is an AC address
AC=0	3	the AC field in the IR is zero
MEM-IDX-IND	4	IR13-17 (indirect bit & index field) /= 0
USER	5	True if the processor is in user mode. (Actually, this bit represents the currently used address space, which may be different from processor mode during XCT mapped)
6?, (EXEC is synonymous with -USER)		
OBUS=0	10	
7?, OBUS<0	11	
JCOND	12	pdp10 conditional decode. i.e. does comparison as specified by the mode bits of the macro-opcode and yields the computed truth value.
OBUS18	13	OBUS bit 18 /= 0.
Q0-35	14	True if low order bit of Q register in 2901's is set
CRY0	15	-carry bit from ALU.
HALF	16	'first part done' flag (bit 04) in PC
BYTE-OVF	17	True if byte pointer in AR will overflow when incremented.

|L

6?

7?

J-CODE bits 09-12.

Jump Code --- microprogram control
May depend on contents of JADR field (for jump address), top element of microsequencer stack, value in "R" register in sequencer, value of jump condition (T or F), or other special hardwired dispatch addresses. Note the PC referred to below is the MICRO-PC.

Mnemonic	Value	Meaning
???	0	PC_0, reset stack
PUSHJ	1	incr. stack ptr, put return address on stack, PC_JADR
LBJUMP	2	T: PC_JADR \/ 1 F: PC_JADR /\ 7776
JUMP	3	T: PC_JADR F: PC_PC+1
	4	(unused)
LOOP	5	R=0: PC_PC+1 R/=0: PC_JADR, R_R-1
POPJ	6	T: PC_STACK, pop stack F: PC_PC+1
JPOP	7	T: PC_JADR, pop stack F: PC_PC+1
	10	(same as 0)
	11	(unused)
DISP	12	Opcode Dispatch. See extended description below.
SDISP	13	Special dispatch: T: PC_OBUS. Note all 16 bits of PC are loaded. (used for switching micro-pages) F: PC_PC+1
	14	(unused)
SLOOP	15	Special Loop. (for division). R=0: PC_PC+1 R/=0: R_R-1 T: PC_JADR \/ 2 F: PC_JADR /\ 7775
CONT	16	(default) PC_PC+1
CONTA		same as CONT, but allows JADR to be specified.
LLOAD	17	R_OBUS (R is a 12-bit internal register in the 2910). PC_PC+1.

|L

Detail for DISP (J-CODE 12)
F2 version:

PC 00-01_JADR 00-01
Mask in low order bits of JADR controls how DISP behaves.

Bit	Does?
04	enables seeing STOP switch
05	enables seeing I/O interrupts
06	enables seeing overflow interrupts
07	enables seeing ECC errors
08	enables seeing indexing
09	enables seeing indirecting
10	Relevant in case JADR 08 or 09 set: PC 08=0 or JADR 10=0: IR used for address calculation PC 08=1 and JADR 10=1: HOLD reg (mem data) used in address calc. Usual indirect and index fields of IR are used, but, for extended addressing, mem data bit 01 is used for indirection, bits 02-05 are used for indexing.

If any of the above conditions are true, don't use instruction code for dispatch.

DISPATCH ADDRESSES: (in order of descending priority)

Address	Condition
3	STOP SWITCH
2	I/O interrupt
5	ECC error
6	arithmetic overflow
7	both ECC and arithmetic overflow
12	indexing, normal addressing (address in IR).
16	indexing, extended addressing (address from HOLD (mem data reg).
10	indirect, normal addressing (address in IR).
14	indirect, extended addressing (address in HOLD (mem data) reg).

If none of the above conditions is true, use (macro-) opcode to generate dispatch address, as follows: the 9 opcode bits (IR 00-08) are shifted left by one bit to generate the 10. low order bits of the dispatch address. (remember the high order 2 bits come from JADR). There are three exception cases:

UUO group
Ignore low order 3 bits of opcode. UUO dispatches are thus on 16-word boundaries.

Group 3 (jumps and skips)
treat just like UUO's.

Group 6 (Test instructions) 600:
Ignore (i.e. zero) opcode bits 06,07.
This generates TEST instruction dispatches on 16-word boundaries, and on 16-word boundaries + 2.

|L

F3 version of DISP (J-CODE 12)

Opcode dispatching is just the same as on the F2, but everything else is different, since the F3 doesn't have extended addressing. In the F3, the STOP switch interrupt is enabled by a special function (SPEC [PC+1-IF&]) rather than by JADR bits. Also, ECC errors and arithmetic overflow are handled by the STOP switch interrupt routine (i.e. there is one interrupt address for all these conditions). In fact, JADR is not used to control interrupt enabling at all in the F2.

DISPATCH ADDRESSES: (in order of descending priority)

Addr	Condition
0	STOP switch interrupt (includes ECC err, OV)
7	I/O interrupt
6	indexing
2	indirecting

MA-STB-FIELD bit 13.

MA register strobe. Loads MA register, either from OBUS, or from PC if SPEC [MA_PC] is set. The usual way of telling the microassembler to load MA is to specify MA in the DEST field, which sets this bit. Loading MA selects the memory output on the memory bus.

Mnemonic	Value	Meaning
NO-MA-STB	0	don't load MA (default)
MA	1	load MA

AR-STB-FIELD bit 14.

AR register strobe. Loads AR register from OBUS. To load AR, can either say AR-STB-FIELD[AR] or DEST[...AR...] (i.e. include AR in DEST field spec).

Mnemonic	Value	Meaning
NO-AR-STB	0	don't load AR (default)
AR	1	load AR

|L

DEST field   bits 15-19.
Specifies which of several registers and memories to load.
Note: if DEST-A-MEM is specified in the SPEC field, the low order 3 bits of the DEST field are used to provide the low order 3 bits of the A-MEM address.
Warning: when using DEST field to specify the low order 3 bits of the A-MEM address (by using SPEC DEST-A-MEM), the left half of the IR is smashed when using A-MEM address 2 or 3 (i.e. the same codes as load the IR when not storing into A-MEM). Also, if DEST bits 00 or 01 are set when writing into A-MEM, the normal DEST is written into.

Mnemonic	Value	Meaning
NO-DEST	0	nothing happens
IR-ADR	1	IR bits 18-35 loaded from OBUS F3 only: generates a signal which tells IR dispatch to ignore non-zero index field
IR-23	2	IR 13-17_memory bus, IR 18-35_OBUS
IR-ALL	3	IR 00-17_memory bus, IR 18-35_OBUS
AC-SEL	4	load AC-CNT register
ROTR	5	load ROT SIZE register
MASKR	6	load MASK SIZE register
I/O-DEV	7	load DEV-ADR register
DEV-ADR	7	
MAP-DISP	10	load MAP DISP register (described in Section 2)
IOD	11	load I/O output data register
CLR-DEV-FROM-INTR	12	Stop selecting address of last interrupting device for hi-order bits of A-MEM address.
HI-ABS-MA	13	Load HI-ABS-MA register from OBUS (described in Sec 2)
CLR-MI-ERR	17	
STRT-WRT	20	start memory write cycle
MEMSTO	21	start memory write cycle load HOLD register from OBUS. Select HOLD on mem bus
HOLD	22	load HOLD register from OBUS. Select HOLD on mem bus
PC	23	load PC from OBUS
CLR-MAP	24	invalidate entire map
STO-MAP	25	map at addr in MA_OBUS 09-26.
CRYOV	26	load all PC flags from OBUS
MAP-EXEC-SR	27	MAP-EXEC-SR_OBUS (described in section 2)
FIXMAC-MAPF-RD	30	Enable map failure trap - read only If MA contains an AC address, load HOLD reg from OBUS.
FIXMAC-MAPF-WRT	31	Enable map failure trap - read/modify/write If MA contains an AC address, load HOLD reg from OBUS.
FIXMAC	32	If MA contains an AC address, load HOLD reg from OBUS.
A-MEM-CNTR&INC	34	Load A-MEM at address whose low 4 bits come from AC CNT reg & increment AC CNT register

```
A-MEM-CNTR        35        Load A-MEM at address  whose low 4 bits come from
                            AC CNT reg
                             Note: AC CNT specifies low order 3 bits
                             of addr, DEV ADR (usually) specifies high
                             order 5 bits.
MUCODE-HI         36        store high order half of microcode mem
MUCODE-LO         37        store low order half of microcode mem
                             (for DEST 36 & 37, address is in AR)

|L
```

CURRENTLY-UNUSED field bit 20.

This bit is available for debugging purposes.

D field   bits 21-24.

Select External Data Source. The data source selected is rotated and masked as described below.

Mnemonic	Value	Meaning
AR	0	
MEM	1	memory bus (= HOLD reg or memory output. see Sec 2)
MASK	2	output of mask generator. doesn't get masked! note: assembler takes argument after MASK which gets stuffed in MASK field
CONST	3	mask field (a literal constant)
PC	4	Selects Flags,,PC. F2 only: PC EXT not read
MA	5	Selects MA, including MA EXT on F2
IOD	6	data from selected I/O device. Note: this is NOT the IOD OUTPUT register !!!
IR	7	
	10-17	Select A-MEM. Low order 3 bits are taken as low order bits of A-MEM address.

|L

ALU
There are 10. bits which control what the ALU does, and to whom.
Three bits (the ALU-D field) control where the ALU output is stored.
Three bits (the ALU field) control the operation performed by the ALU.
Three bits (the SOURCE field) control the selection of inputs to the
ALU. One other bit determines the carry-in to the ALU (the CARRY-IN field).

In general, one need only specify the ALU operation field to Dyer's assembler, which will set the SOURCE and CARRY fields to values implied by the operation mnemonic. The ALU-D field must be specified explicitly. It is possible to include ALU-D specifications in the argument to the DEST field.

ALU-D field bits 25.-27.
Destination of ALU operation. Note the OBUS also gets the result of the operation, except for code 2.

'B' refers to the 2901 internal memory at address B (see ACSEL field).
'Q' refers to the Q register in the 2901's.

Mnemonic	Value	Meaning
Q	0	result goes to Q register in 2901
NONE	1	result not stored
O-AC	2	"old AC". ALU results stored at address B in internal memory. OBUS gets contents of memory at address A.
AC	3	ALU results stored at address B. OBUS also gets ALU results!
D4 B-RIGHT&Q-RIGHT	4	B gets alu results shifted right by 1. Q also shifted right by 1.
D5 B-RIGHT	5	B gets alu results shifted right by 1.
D6 B-LEFT&Q-LEFT	6	B gets alu results shifted left by 1. Q also shifted left by 1.
D7 B-LEFT	7	B gets alu results shifted left by 1.

|L

The following description does not use assembler mnemonics, which are best learnec from the microcode definition file. Instead, the actual 2901 functions are described.

ALU field bits 28.-30.

"R" and "S" are the two inputs to the ALU. R may be selected from 0, memory at address A, or an external source (see D field). S may be selected from 0, memory at either address A or B, or the Q register. (See the SOURCE field).

Value	Meaning
0	S+R
1	S-R
2	R-S
3	R OR S
4	R AND S
5	-R AND S
6	R XOR S
7	R EQV S

SOURCE field bits 31.-33. (depends on dest + Q if D4-D7 &
Selects inputs to ALU

When ALU-D < 4, the following definitions hold:

Mnemonic	Value	R	S
A,B	0	A	B
A,Q	1	A	Q
0,B	2	0	B
0,Q	3	0	Q
D,A	4	EXT	A
0,A	5	0	A
D,0	6	EXT	0
D,Q	7	EXT	Q

the shifts to B or B&Q

When ALU-D >= 4, things get more complicated, and the SOURCE field is instead referred to as COND-SOURCE. This is used for Multiply and Divide. Note that with ALU-D >= 4, the source depends partly on the low order bit of the Q register.

		if L.O.Q.=0		if L.O.Q.=1	
Mnemonic	Value	R	S	R	S
D,A	0	EXT	A	EXT	A
0,A	1	0	A	0	A
D,0	2	EXT	0	EXT	0
D,Q	3	EXT	Q	EXT	Q
0,A-OR-D,A	D4 4	0	A	EXT	A
	5	0	A	EXT	A
D,Q-OR-D,0	6	EXT	Q	EXT	0
	7	EXT	Q	EXT	0

CARRY-IN field bit 34.

Carry In to the ALU. This bit is set implicitly by using various ALU function mnemonics which imply its value. It may be set explicitly if so desired:

Mnemonic	Value	Meaning
CARRY	1	set carry-in
NO-CARRY	0	(obvious!)

ACSEL field bits 35.-36.

Selects sources for addresses to internal memory in the 2901's.

Mnemonic	Value	Meaning
REG	0	both A & B addresses from AC CNT register
MA	1	A addr from index field of IR, B addr from MA bits 32.-35.
AC	2	Both A & B addresses from AC field of IR
AC+1	3	Both A & B addresses from (IR AC)+1

ROT-SEL field bit 37.

If this bit is on, rotation count comes from ROT SIZE register instead of the ROT field.

ROT field bits 38-43.

Rotation count. Specifies how much to rotate the data specified by the D (external source) field. ROT [nn] specifies a rotation by the (octal) number nn. ROT [R] magically sets the ROT-SEL bit.

MASK-SEL field bit 44.

If this bit is on, the mask size comes from the MASK SIZE register instead of the MASK field.

MASK field bits 45.-50.

Mask size. The rotated data from the external data source, as specified by the D field and the ROT field, is anded with 2|<mask-size> - 1. The result of the masker goes to the ALU external input. To set the mask size, use MASK [nn]. MASK [R] sets the MASK-SEL bit instead. There are exceptions to the above logic, which are detailed in the general description below. (far below)...

PAR field   bit 51.

Parity bit. Is required to generate EVEN parity for the microinstruction word. Currently, this is handled by the microcode loader.
NOT the one in ROM !

|L

SPEC field bits 52-55 (4 bits)
This field specifies a "special function"

Mnemonic	Value	Meaning
NO-SPEC	0	no special function
LEFT	1	mask = -1,,0
LEFT&MA_PC	2	load MA from PC (use with STB MA) mask = -1,,0
MA_PC	3	load MA from PC (use with STB MA)
CRYOV&MA_PC	4	load MA from PC (use with STB MA) set carry & overflow flags from alu results
CRYOV	5	set carry & overflow flags from alu results
PC+1-IF&	6	F3 version: incr. PC if dispatching on opcode and not on int, idx, or ind. Enable STOP switch interrupt
???	6	F2 version: MA EXT_PC EXT, MA (low order 18 bits)_OBUS
PC+1-IF	7	increment PC if dispatching on opcode and not on int, idx, or ind.
PC+1	10	increment PC
CLR-HALF	11	clear 'FIRST PART DONE' flag
A-MEM-APR	12	force high order 5 bits of AMEM addr to be 0. i.e. select device 0
A-MEM-APR&DEST-A-MEM	13	force high order 5 bits of AMEM addr to be 0. store OBUS into AMEM at address specified by low order 3 bits of DEST field i.e. write into CPU's part of AMEM
DEST-A-MEM	14	store OBUS into AMEM at address specified by current DEV ADR and by low order 3 bits of DEST field
IOB-IN	16	request i/o bus input cycle.
IOB-OUT	17	request i/o bus output cycle. next micro-cycle must have MAPF field = device subselect. on input, must read data from dev.

MAPF field bits 56-59.

Map Fault Dispatch Ccce.
Provides certain bits of the dispatch address for map faults.
(See description of MAP DISP register below).
Also, provides I/O device subselect (i.e. function code) during
I/O operations (see description of I/O below).

JADR field bits 60-71. (12 B)

Jump Address. Also used for dispatch enable bits in the F2.

|L

Section 2. General Description

REGISTERS

IR (macro-) instruction register. Left half loadable from memory bus (which is either the HOLD register or the memory output). Right half loadable from the OBUS only. To load whole IR, use DEST [IR-ALL]. To load IR13-35, use DEST[IR-23]. To load IR18-35, use DEST[IR-ADR]. Note (on the F3), DEST [IR-ADR] has the side effect of disabling the IR dispatch from taking the indexing trap, even though it does not zero the index field in the IR. IR is readable as external data with D[IR].

PC (Macro-) Program Counter. Loadable from OBUS with DEST [PC]. Readable as external data with D[PC]. When using D [PC], the flags are read back along with the PC. Incremented with SPEC [PC+1], SPEC [PC+1-IF], or (on the F3) SPEC [PC+1-IF&] (see description of SPEC field). In the F2, the PC has a 6 bit extension. The extension is loaded along with the rest of the PC with DEST [PC].

HOLD Memory Hold Register. Data written to main memory always comes from HOLD. HOLD is loadable from the OBUS, with DEST [HOLD]. HOLD is conditionally loaded from OBUS with DEST [FIXMAC] etc. if MA contains an AC address (MA <= 16.). DEST [MEMSTO] loads HOLD from OBUS while also starting a write cycle. (Note it is the NEW data in HOLD which gets stored.) HOLD is directly readable as external data with D [MEM].

MA Memory Address register. Used as memory address for all references to main (macro-) memory. Low order bits can select AC address in 2901 internal memory. Loadable from OBUS or directly from PC. SPEC[MA_PC] or SPEC[CRYOV&MA_PC] select PC instead of OBUS. To load, use STB MA bit (can say MA in DEST field). Readable as external data, with D[MA]. In the F2, the MA has a six bit extension on the high order end (for extended virtual addressing!). If PC 08=0, MA EXT is cleared whenever MA is loaded. IF PC 08=1, MA EXT is loaded from the appropriate bits of the OBUS or from the PC extension. The MA extension is read back with the rest of the MA with D[MA].

HI-ABS-MA

This is (another) extension to the MA, which exists on both the F2 and the F3. It is used when the map is off, and not otherwise. It is loaded with DEST = 13 (DEST[HI-ABS-MA]), and can't be read back. This is a 4-bit register. Data with 22 bit addresses can be read with the map off, but code cannot be run out of the extended address space provided by HI-ABS-MA.

AR Used as address for storing into Microcode Memory. Loadable from OBUS with STB AR bit. (can say AR in DEST field). Readable as external data, with D[AR]. The jump condition BYTE-OVF is determined from the data in AR.

AC CNT A 4-bit counter. May be selected to address internal memory in the 2901's. Can be used as low order 4 bits of write address into A-MEM.

ROT SIZE

Used to control rotator in external data path. Loadable from OBUS using DEST[ROTR]. Selected with ROT-SEL bit by saying ROT[R].

MASK SIZE

Used to control mask generator in external data path. Loadable from OBUS using DEST[MASKR]. Selected with MASK-SEL bit by saying MASK[R].

MAP DISP

MAP DISP is a 6-bit register. Loadable from OBUS with DEST [MAP-DISP]. Specifies high order bits of map fault dispatching:

F3: map fail dispatch addr = ((MAP DISP)*2000) + 100 + (MAPF*4).

F2: read or read-modify-write cycle:
dispatch addr = ((MAP DISP*2000) + 500 + (MAPF*4)

write cycle:
dispatch addr = ((MAP DISP*2000) + 600 + (MAPF*4)

IOD I/O Output Data register. Used for SENDING data to devices on the I/O bus. Load with DEST [IOD].

DEV ADR

This is the Device Address register. It is loaded from the OBUS with DEST [DEV-ADR]. It is used to generate the high order 5 bits of the address to A-MEM (it is one of several options here). DEV ADR is also connected to the I/O bus so that devices there can know if they are being talked to.

Q An internal register in the 2901 chips.

|L

A-MEM

This is a 36. bit x 256. word memory which is used primarily for communication with I/O devices. The memory is (logically) divided into 8-word blocks, each of which is associated with a given I/O device.

The data written into the A-MEM comes only from the O-BUS, and is only written when SPEC[DEST-A-MEM] (14) or SPEC[A-MEM-APR&DEST-A-MEM] (13) are specified.

Data is read from the A-MEM by setting the D field (external source) to a value in the range 10-17.

The high order 5 bits of the 8 bit A-MEM address come from one of three sources:

The DEV-ADR register is used by default.
This register is loadable from the OBUS with DEST [DEV-ADR] (7).

Address of last interrupting device is used when processing an interrupt. This option is selected by the interrupt dispatch during a DISP operation (see description of instruction dispatching).

0 is used when SPEC = 12 (A-MEM-APR) or SPEC = 13 (A-MEM-APR&DEST-A-MEM). By convention the 0th block of A-MEM is associated with the CPU (device APR).

Note: To stop selecting the address of the last interrupting device, it is necessary to execute a micro-instruction with DEST = 12 (CLR-DEV-FROM-INTR).

The low order bits of the A-MEM address come from one of three sources:

When DEST[A-MEM-CNTR] (35) or DEST[A-MEM-CNTR&INC] (34) are used, the low order 4 bits of the A-MEM address come from the AC CNT register. In the second case mentioned, the AC CNT register is incremented after the memory reference. NOTE: the AC-CNT register can be used for addressing A-MEM for either reading or writing. Also, the AC CNT register specifies the low order 4, not 3, bits of address. The highest order bit supercedes the low order bit of the DEV-ADR reg or whatever other source of high order address bits is being used.

When writing into A-MEM, the low order address bits come from the low order 3 bits of the DEST field.
Do not use DEST field values greater than 7, since if you do, the normal DEST will be smashed as well as the A-MEM being written.

When reading A-MEM, the low order 3 address bits come from the low order 3 bits of the D (external source) field.
D field values 10-17 select A-MEM addresses whose low order bits are 0-7 respectively.

|L

External Data, Masking, and Rotation

As described above, the D field selects one of several data sources. This data is then rotated by the amount in the ROT field, unless the ROT-SEL bit is set, in which case the amount in the ROT SIZE register is used. The ROT SIZE register is loadable from the OBUS with DEST [ROTR] (5).

The rotated quantity is then ANDed with the output of the mask generator. The mask generator takes as input the MASK field, unless the MASK-SEL bit is set, in which case it takes the contents of the MASK SIZE register. It converts this binary quantity to a right-justified unary mask. There are two exceptions to this. If SPEC LEFT (SPEC = 1), the mask is forced to -1,,0. If D = CONST, or D = MASK, the masking is not done. The MASK SIZE register is loadable from the OBUS with DEST = 6 (MASKR).

The result of the selection-rotation-masking operation becomes the D (external) input to the ALU. Furthermore, this data doesn't go anywhere else.

Unless otherwise specified, the default ALU mode and source selection is to put the 'D BUS' (result of selection, rotation, shifting) onto the OBUS. (This is an artifact of Dyer's micro-assembler!).

ALU

The ALU is made of 2901's. The 2901 has 8 different functions of 2 inputs. The inputs can come from an internal memory, an internal register, or an external source.

There is a 16. word memory in the 2901's which is used as the accumulators in the macro-machine. The memory is structured so that two different locations can be read at the same time (and fed to different ALU inputs). One of the two addresses can be used to write into the memory. There is also another register in the 2901's called Q, which can be loaded from the ALU output and used as input. When the ALU output is stored into the memory, it can be shifted one place right or left. It is also possible to shift the contents of Q.

The ALU output is in general placed on the OBUS. (There is an exception...see description of ALU-D field).

The ACSEL field controls the addresses to the internal memory in the 2901's, as described above.

MEMORY and how to use it

As previously noted, the HOLD register is the source for all cata written to or read from memory. The MA register is always the address register.

Memory data from the last memory operation is selected with D [MEM]. Whenever the HOLD register is loaded (with DEST [HOLD] or DEST [MEMSTO]), the HOLD register is put on the memory bus, and this is what is seen with D [MEM]. Whenever MA is loaded (and thus a read cycle begun), the memory contents themselves are placed on the memory bus. Note that if DEST [HOLD] is done, D [MEM] sees the HOLD register regardless of the contents of memory. If DEST [HOLD] is done, several read operations can be done without affecting the HOLD register, and then a DEST [STRT-WRT] will write the contents of the HOLD register.

Reading

Loading the MA register starts a read cycle. Any time later, an instruction with valid MAPF field and DEST [FIXMAC-MAPF-RD] must be done to take care of the case where an AC was addressed or there was a map failure. If there was a map failure, the next instruction executed is taken from the location as specified in the description of the MAP DISP register. If a write to the same address is being planned, use DEST [FIXMAC-MAPF-WRT] instead of DEST [FIXMAC-MAPF-RD], since this will also take the map failure trap if write prevent is on for the page.

Writing

A write cycle begins during an instruction with DEST [MEMSTO] or DEST [STRT-WRT]. HOLD or MA may be loaded as late as the instruction including the command to start the write cycle. The next instruction executed must include a valid MAPF field, since the write automatically enables the map fail dispatch.

I/O operations

Input: An instruction including SPEC [IOB-IN] requests a bus cycle. DEV-ADR may be loadec as late as this instruction. The next instruction executed should have MAPF set to the device subselect code for the required operation. Also, during this instruction D[IOD] should be set to read the requestec data. CYLEN should be set to IOB-IN to allow enough time.

Output: An instructicn including SPEC [IOB-OUT] requests a bus cycle. DEV-ADR or IOD may be loaded as late as this instruction. The next instruction must have MAPF set to the device subselect code for the required operation. The contents of IOD are sent to the device. CYLEN should be set tc IOB-OUT.

|L

Section 3.

Instruction Timing Rules --- how to set up the CYLEN field.

Short cycle (350 ns) can be used only if J-CODE is CONTINUE, MI DEST 20 is off this cycle anc last cycle, and no mem write was started last cycle or cycle before last.

In the following table, the numerical entries are of course the necessary nominal cycle length in nanoseconds. "int" and "ext" refer to whether the source data to the ALU is internal to the ALU (AC's or the Q register) or an external source.

		ALU OPERATION:			
		LOGICAL		ARITHMETIC	
J-CODE	COND	int	ext	int	ext
CONT,LOOP,LLOAD,JUMP-ZERO	irrelevant	400	400	400	400
JUMP,PUSHJ,POPJ,JPOP	Fast (*)	400	400	400	400
"	Slow (**)	450	550	500	600
"	JCOND	500	550	550	600
LBJUMP, SLOOP	Fast (*)	400	400	400	400
"	Slow (**)	500	550	550	600
"	JCOND	500	600	550	650
SDISP	Fast (*)	450	500	500	550
"	Slow (**)	450	550	500	600
"	JCOND	500	550	550	600
DISP	TRUE	500	500	500	500

(*) Fast conditions are: TRUE, USER, AC=0, MA-AC, INTRPT, BYTE-OVF, HALF
(**) Slow conditions are: OBUS=0, OBUS<0, OBUS18, CRYO

>>>>>>> What about MEM-IDX-IND, Q0-35 ???