TECHNICAL NOTES ON THE EEC-IV MCU

INTRODUCTION

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

DISCLAIMER

(The information supplied herein was taken from technical and sales literature, e-mail archives, news groups, and wherever else it could be found. If it helps you, great! Please return the favor by sharing what you learn with me and others.)

None of this data is guaranteed accurate! **USE AT YOUR OWN RISK!** It is a collection of technical info, opinions and guesses. You can contribute to the effort by letting me know what you learn so that it can be added to and corrected.

PURPOSE

This data has been collected and compiled to help you decipher the EEC-IV inner workings. The EEC MCU probably controls one or more vehicles you own, plus it contains all the components necessary to build an EFI system for any vehicle -- if only we could program and modify it. That is the purpose -- to uncloak the EEC-IV so that we can play with what we bought! Much of the empirical data in this document is specific to the A9L EEC computer. That is the model MotorSport SVO sells for conversion of Mustangs from SD to MAF, and it was stock on 5-speed manual transmission Mustangs from 1989 to 1993, so consensus was reached to pursue this one configuration until it was understood rather than divide our efforts chasing many.

Beginning with contacts on the Fordnatics and DIY-EFI mailing lists, the discussion and research into the EEC PCM was soon split off into its own e-mail group, the EEC-IV Mailing List. To participate in the ongoing work of uncloaking the EEC, join the EEC mail list on the internet.

Information on EFI Mailing Lists can be found at

EEC-IV	http://eelink.umich.edu/~p-nowak/eec-efi/EEC-Mailadddrop.html to subscribe, send mail (your mail message can be empty) to: eec-subscribe@eelink.net
DIY-EFI	http://efi332.eng.ohio-state.edu/ to subscribe, send an email to Majordomo@efi332.eng.ohio-state.edu with the following in the body of your message: subscribe diy-efi [your email address *only* if different than your "From" address]
Fordnatics	http://www.neosoft.com/internet/paml/groups.F/fordnatics

Table 1: EFI Related Mailing Lists

To subscribe, send email to fordnatics-request@lists.best.com and in the body of the message, put the word “ subscribe”

Table 1: EFI Related Mailing Lists

OVERVIEW

The EEC-IV design began in 1978 and was first introduced in 1983 in the 1.6L Escort, Lynx, EXP and LN7 cars. It has gone through several major physical changes, the earliest using a fairly simple two board design with through hole soldered components while the last are more current in technology, showing extensive use of surface mount components and a much more finished and complex appearance. In between, there appears to be a variety of mother/daughter board and other designs. Still, they are all called EEC-IV, although somewhere in its life there was a Ford P/N generational change.

The reader is referred to the SAE paper #820900, noted in the reference section at the end of this document, for a much more detailed description of the design goals and operation of the EEC-IV MCU.

One person wrote: "The processor used is the 8065 along with several supporting peripheral chips like the DUCE chip which can provide up to 8 PWM outputs and the DARC chip which has 6 channels of timer capture inputs." (Is he talking about the EEC-V here ?)

The EEC module is rated to 80^{o}C (185^{o}F) continuous, 100^{o}C intermittent, so it will be much happier and live longer in the passenger compartment. Some of the later generation 15 and 18 MHz Motorola 8061 processors have a bus loading/edge timing sensitivity that only gets worse at high temperature, so it’s best to keep the EEC in a more hospitable environment. Additionally, mounting the EEC in the passenger compartment will give you better access to the J3 test port, which is where you’ll be plugging in an after-market module or any test / modification device.

The J3 test port on the side of the ECU box is for developers to plug into -- this is how the after-market chipmakers and others get into the box. The test connector has the micro-controller’s multiplexed address/data bus signals on it. It also, very conveniently, has a PROM disable signal. So the chip makers design something that hangs off that connector, disables the computer’s PROM, and substitutes its own PROM in its place.

The reader is encouraged to investigate the Intel 8096 (MCS-96) literature for greater insight into the 8061 processor used in the EEC. One document available from their web site is "27006102.pdf" entitled "APPLICATION NOTE; AP-248; Using The 8096"; Order Number: 270061-002. There are many other related documents available from Intel -- including the use of the A-D converter, the implementation of "fuzzy logic", instrumenting and controlling automotive applications, and other topics.

Also, this author has posted a synopsis of the 8096 pinouts and instruction set which is available at several web sites.

ACKNOWLEDGEMENTS / CONTRIBUTORS

Gotta love this job - I collect all the work other people have done and then get to take the credit for it! Someone sent this to me via e-mail and I thought it appropriate here: “To steal ideas from one person is plagiarism; to steal from many is research”.

Below is a list of the people (and/or other sources) that have either given me information I've used here or have provided insight or help - they're the ones that know about this subject. Correction - a consensus was reached to omit references to any people - so, if you contributed something or are quoted in this document and want to be acknowledged, please let me know and I'll be sure you're mentioned in the next rev.

(I've been doing this so long, have talked with so many people, visited so many web sites and read so much literature that I guarantee many people and sources are omitted. It's unintentional. If you've been left out or you see something or someone that should be acknowledged here, please let me know right away.)

* [the names were here]

* Intel MCS-96 literature

* Intel Application notes

* AP-248; Using the 8096

* "How to Understand, Service, and Modify Ford Fuel Injection and Electronic Engine Control", by Charles O. Probst

* SAE paper #820900, "EEC-IV Tomorrow's Electronic Engine Controls Today", David Hagen & Dennis Wilkie, Ford Motor Co., Dearborn, MI

Date	Name(s)	Notes
9-26-96	Subject: Intel 8061 :: EEC-IV	e-mail
2-7-97	Subject: EEC-IV notes	e-mail
2-14-1997	Subject: Ignition Writeup	e-mail
4-22-1997	8096_CPU.DOC	8096 op-codes, etc
11-26-1997	112601.pdf	
9-3-1998	Eectch98-Intro.pdf rev 2 Eectch98-Part1.pdf Eectch98-Part2.pdf Eectch98-Part3.pdf Eectch98-Part4.pdf Eectch98-Part5.pdf	
9- -1998	Eectch98.pdf rev 3	conversion to FrameMaker 5.5

Table 2: Document Revision History

Table of contents

Figure: 8061 Microprocessor

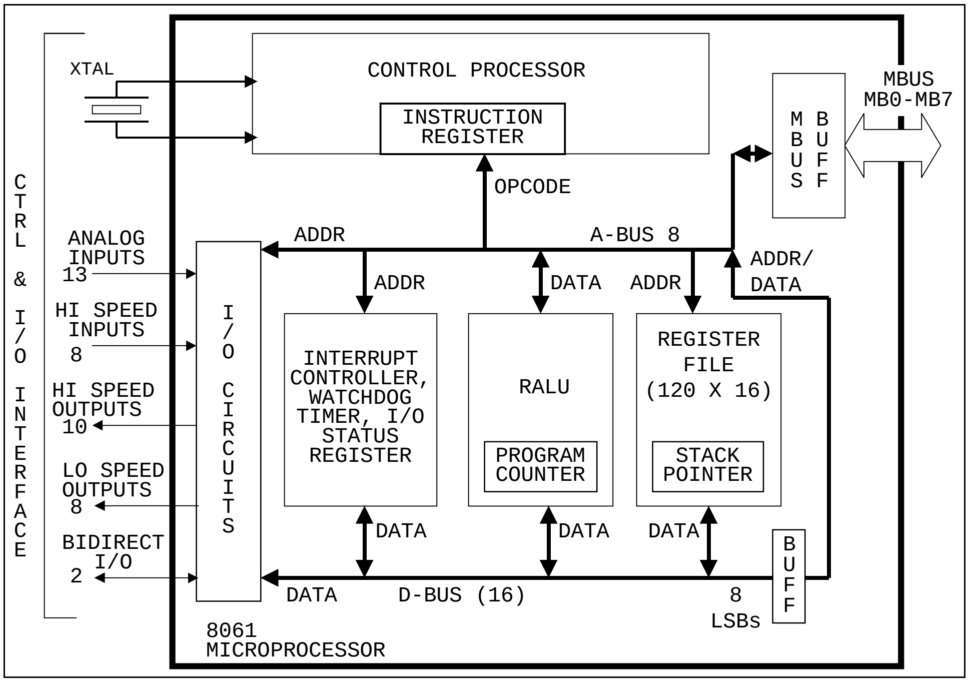
Figure: 8061 Major Functional Units

HARDWARE

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

THE MICROPROCESSOR / MEMORY CHIP SET:

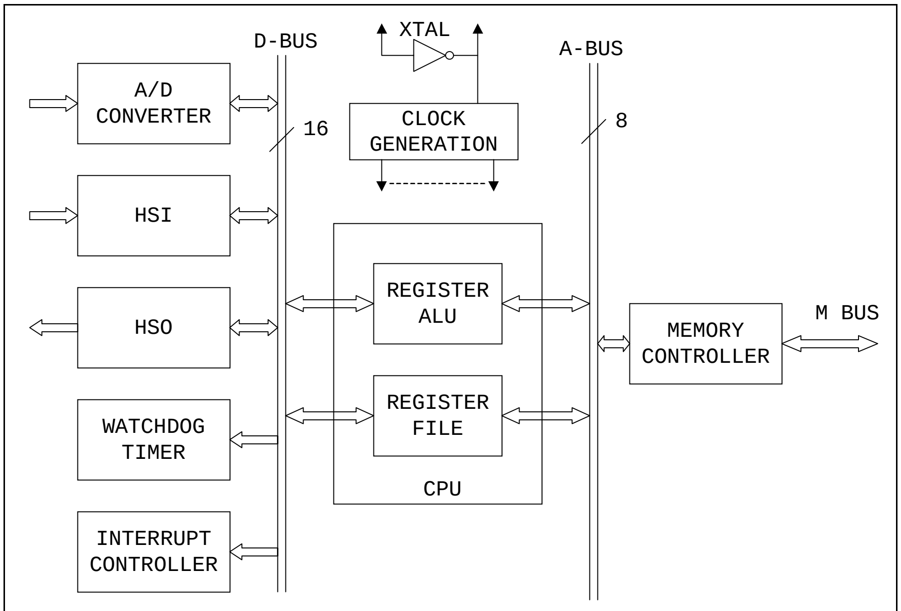
The micro-controller is an Intel 8061, a close cousin to the Intel 8096. It is supplied by three manufacturers: Intel, Toshiba (6127) and Motorola, though the Motorola units are said to slip spec a little and differ in their timing slightly from the others. It was originally (in 1978) intended to operate in conjunction with an 8361 ROM.



There are some major differences between the 8061 and 8096 (e.g. pinouts, bus layout, etc.), but most of the code appears transferable.

It is organized internally as a 16-bit machine with a double bus structure consisting of CPU, memory controller, clock generator, I/O and co-processors, A/D converter, watchdog timer and interrupt controller.

The high speed hardware / register structure is a design by Ford engineers to simplify the processing of digital I/O signals. Ford chose to design a custom microprocessor - memory combination and the 8061 and 8361 were the result. Those two chips form a two-chip microcomputer.



8061 Major Functional Units

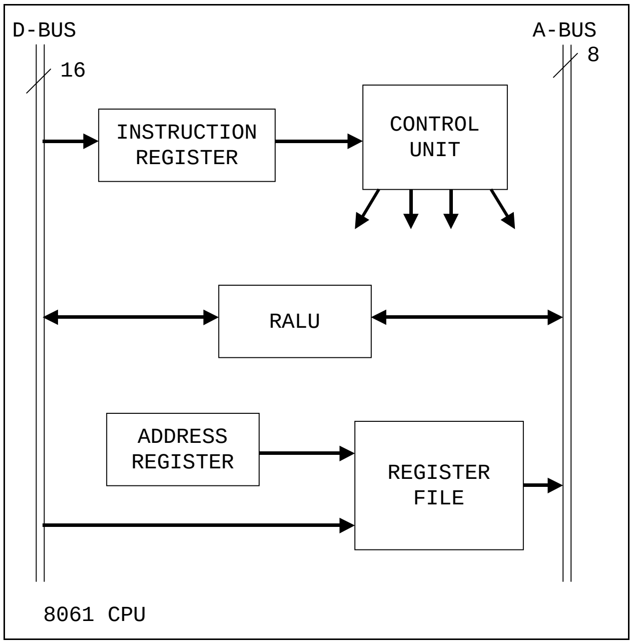
There were several design goals for this custom microprocessor:

1. An I/O intensive circuit with hi-speed digital I/O capability.
2. A fast, on-chip, multi-channel A/D converter.
3. Hardware multiply and divide.
4. Multi-level,

prioritized interrupts.

5. Variable data types (bit, byte, word & double word).
6. A watchdog timer.
7. A powerful yet “regular” software architecture.
8. A large memory address space with minimum off-chip memory access time.

The 8061 microcomputer chip features a CPU, 256 bytes of RAM, an A/D converter and independent co-processor circuitry to expedite digital signal I/O handling. There are 13 analog lines, 8 hi-speed digital inputs, 10 hi-speed digital outputs, 8 lo-speed digital outputs and 2 bi-directional I/O lines, making a total of 41 I/O lines on the CPU chip. The A/D converter is a 13-channel, 10-bit successive approximation unit.



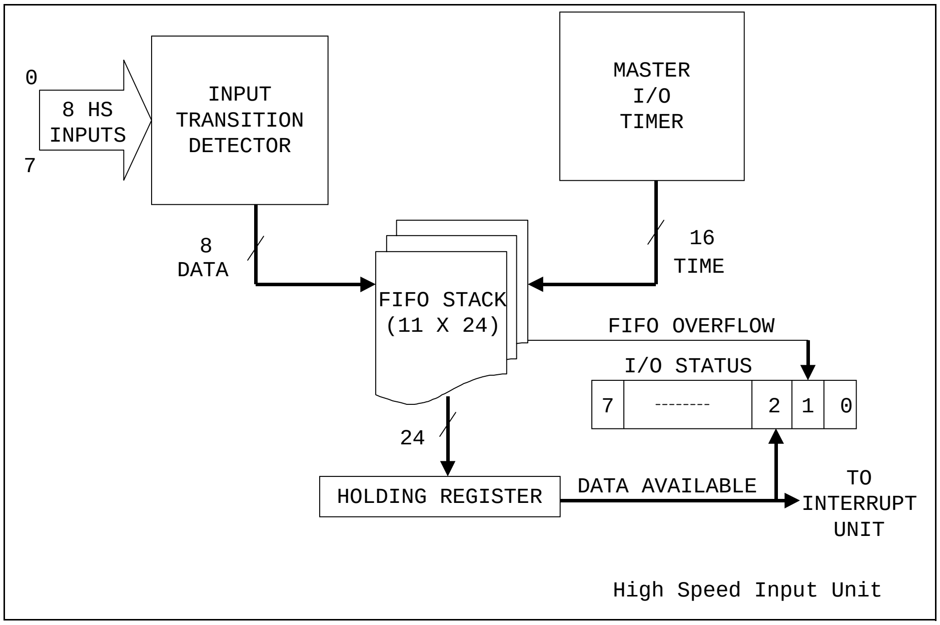
8061 CPU

The internal 256 bytes of RAM in the 8061 can be referenced as bytes, words or double words, allowing frequently used variables to be stored on-chip for faster access.

The two high speed co-processors on the 8061 (HSI and HSO) were implemented to reduce signal processing overhead on the CPU. An 11-deep FIFO for the high speed input (HSI) and a 12-slot content addressable memory (CAM) for the high speed output (HSO) are used. Operation of both HSI & HSO are synchronized with an internal master I/O timer which is clocked every 2.4 microseconds (15 MHz crystal).

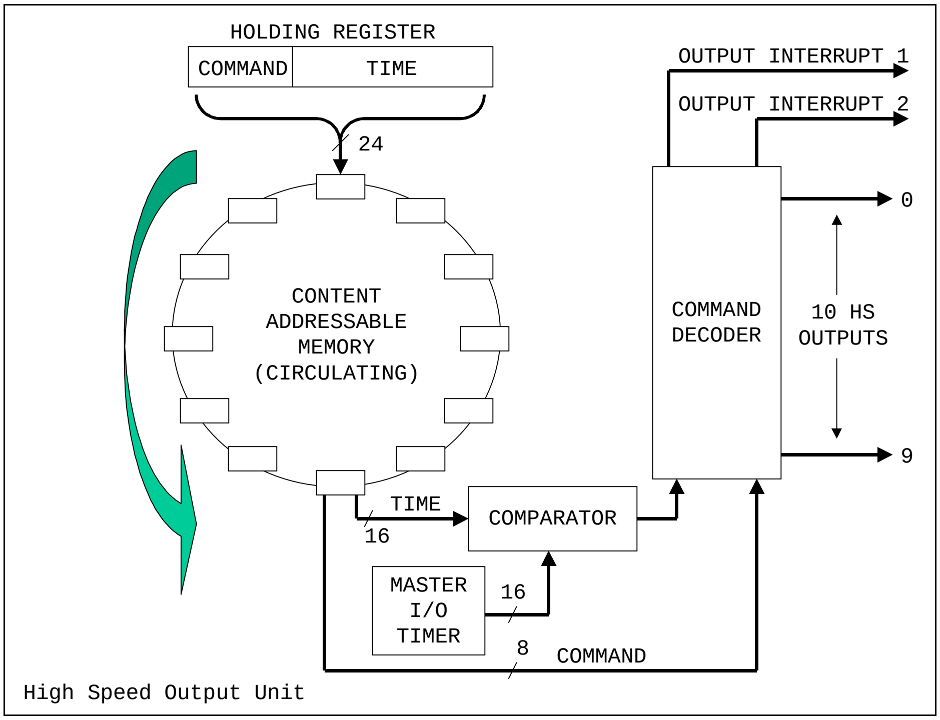
The HSI looks for transitions on its input lines and records (1) the time, from the master I/O timer, and (2) the transition. It can be programmed to look at selected inputs for positive and negative transitions and can be programmed to generate an interrupt to the CPU when the first entry is made into the FIFO or when the next entry would cause the FIFO to overflow.

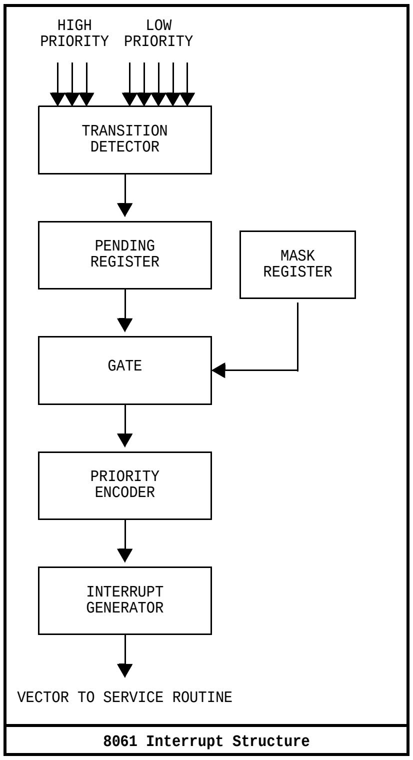
The HSO can be programmed to generate transitions on any of its output lines at specified times. HSO commands are stored in one of the twelve CAM registers, which are 24 bits wide. Of the 24 bits in each register, 16 specify the time the action is to occur, and 8 specify the action(s). The CAM file rotates one position per state time, so it takes 12 state events for the holding buffer to access all 12 registers. Therefore the time resolution of the HSO unit is 12 state times or 2.4 microseconds if a 15 MHz crystal is used.



High Speed Input Unit

Two “novel” architectural concepts were used. One, address and data is multi-





8061 Interrupt Structure

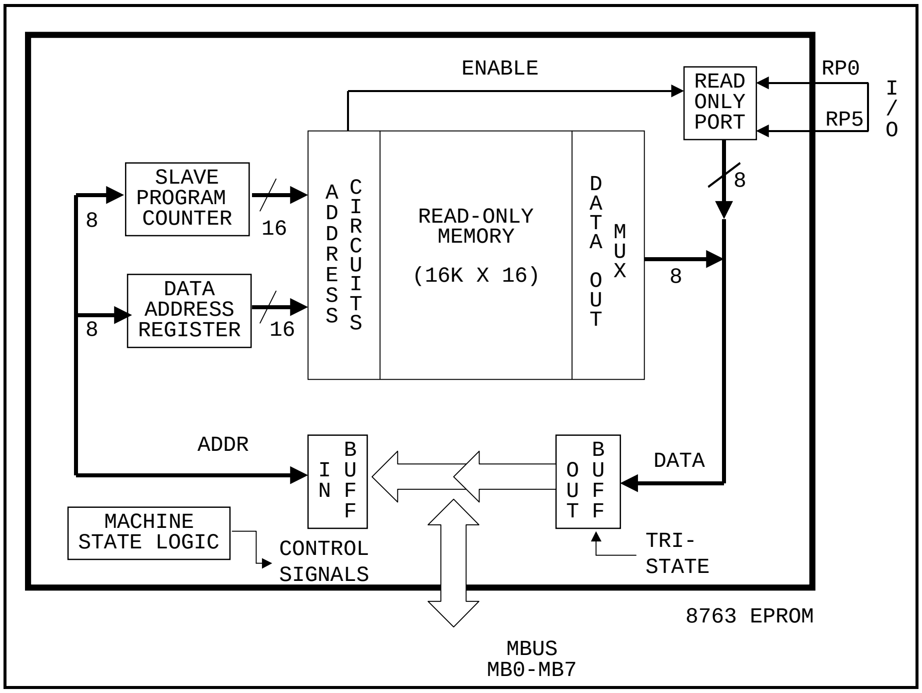
plexed on the external memory bus (M-bus) to minimize the number of memory interface pins. This isn't novel, but the collateral addition of slave program counters on the memory chips is. Second, “a large (256 bytes of RAM) general purpose register file was designed into the 8061". It can be referenced as byte, word or double word allowing storing and accessing data directly on-chip.

The 8061 is a double-bus structure machine with CPU, memory controller, clock, I/O with co-processors, A/D converter, watchdog timer, and an interrupt controller.

The CPU consists of the register file, the register-arithmetic logic unit (RALU), and a control unit. Note that the RALU does not use an accumulator but operates directly on any register in the register file, resulting in code length and execution speed improvements. The control unit consists of the instruction register and associated circuitry which decodes the instructions and generates the correct sequence of internal control signals to execute instructions.

The clock generator in the 8061 divides the crystal frequency, internally, by three to provide a duty cycle of 33%. The clock signal period, called one state time, equals three oscillator periods.

A watchdog timer is incremented every state time. It is a 16-bit counter that reinitializes the system when it overflows to provide a means of recovering from a

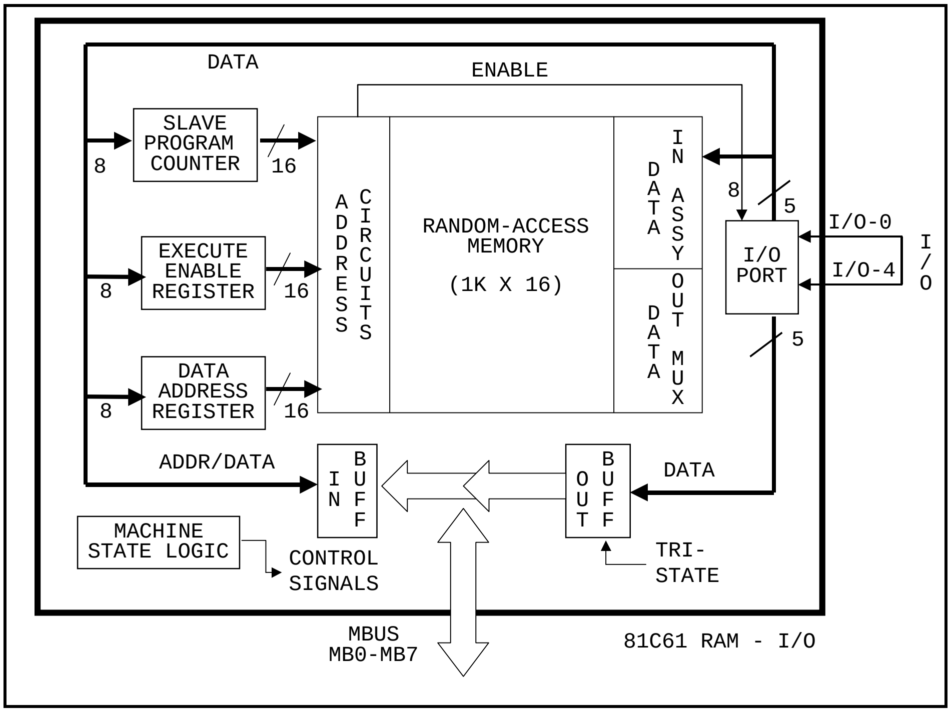


software fault. The user must periodically reset the watchdog timer to prevent register overflow and subsequent re-start.

There are 8 interrupt sources in the 8061. A positive transition from any one of the sources sets a corresponding bit in the pending register. A programmable mask register determines if the particular interrupt will be recognized or not. Interrupts can occur at any time and simultaneous interrupts are accepted. Conflicts are resolved with a two-level sequential priority hierarchy which establishes the order of servicing. A corresponding vector automatically identifies the location of each interrupt service program. A software stack, which can be created anywhere in memory, can be used for temporary storage of important program data (e.g. the PC and PSW) during execution of interrupt service routines.

The 8061 can address up to 64k bytes of memory, supports bit, byte, word and double-word data types with six addressing modes and has eleven instruction categories defined. The assembly language programmer can create very fast, compact programs by using the direct addressing mode and careful movement of data between external memory and the register file.

The 8361 ROM chip contains 8k bytes of program memory plus 128 bytes of additional RAM. Data transfer between the 8763 ROM and the 8061 is controlled by the memory controller in the 8061. Addresses for instruction fetches from the ROM are maintained in a slave PC in the 8061 memory controller and in a corresponding counter in the 8763. The slave PC functions like a traditional PC, being automatically incremented after each fetch and updated whenever the CPU executes a program jump. The counter in the ROM is independent of the slave PC but is identical to it. Addresses are transmitted on the M-bus from the slave PC to the ROM under two conditions, when the address is initialized at the start of program execution or when a program jump occurs. The slave PC concept eliminates the need to send an address to external memory for each instruction -- that only being necessary when a branch occurs or at program initiation.



The 8061 is an 8096 with a few extra instructions added. One is a very powerful conditional jump to complement the high speed I/O units. This instruction, the jump on bit equals zero, is used to test any one of the eight bits of a given byte and jump if the bit equals zero (is this the JBC/JNB command?). Other conditional jumps were added to avoid extensive data shifts. With a 15 MHz input frequency, the 8061 can perform a 16-bit addition in 0.8 microseconds and a 16 x 16 bit multiply or a 32/16 bit divide in 5.2 microseconds (using the hardware multiply and divide feature). For typical applications, based on a normal instruction mix, instruction execution times average 1 to 2 microseconds. It seems to have the same functional pins as the 8096, but it's in a custom package, so the pinout is different. Most of the signals should be able to be found with a scope or logic analyzer. The 8096 has a multiplexed address/data bus. The address/data bus signals are on the service port connector (J3) along with a few others, possibly including the address latch enable, read strobe, write strobe, and EPROM disable.

There are two hardware versions of the 8061 chip. One is a 40 pin DIP, with reduced I/O and the other is a square LCC 68 pin package with all the functions implemented.

The multiplexed M-bus scheme used on the 8061 is not new, but the slave program counter used on the 8763 is. It appears that the address / multiplexing scheme is similar to that of the 8085 which has AD0 .. AD7 and then A8 .. A15 so the 8085 “latches” the address information A7:0, and maintains A8:15 while it is using AD0 .. AD7 as D7:0 ....

It is speculated that the only internal EEC memory you could possibly get would be the FLASH memory, if it was ever used. Look for a PLCC package, probably 32 pin, that should be a standard 28F010 12V write, erase/5V read Flash available from TI, Intel, AMD, and others. Might be labeled 81C65 or 81C62.

LEGEND			
ADDR	ADDRESS	I/O	INPUT/OUTPUT
ASSY	ASSEMBLY	LO	LOW
A-BUS	ADDRESS BUS	LSB	LEAST SIGNIFICANT BIT
BIDIRECT	BIDIRECTIONAL	MBus	MEMORY BUS
BUFF	BUFFER	EPROM	ERASABLE READ-ONLY MEMORY
CTRL	CONTROL	MUX	MULTIPLEXER
D-BUS	DATA BUS	RAM	RANDOM ACCESS MEMORY
HI	HIGH	RPn	READ-ONLY PORT INPUT

Table 3: EEC Chipset Nomenclature Legend

8061 CPU (IC-1)					
1		unused	35		A4
2		unused	36		A5
3		/RESET	37		A7
4		HSO9	38		A6
5		HSO8	39		A2
6		HSO7	40		A0
7		HSO6	41		A1
8		HSO5	42		A3
9		HSO4	43		AVCC
10		HSO3	44		VSS
11		HSO2	45		NC
12		HSO1	46		VBB
13		HSO0	47		HSI0
14	GND	VSS	48		HSI1
15		xtal2	49		HSI2
16		xtal1	50		HSI3
17		LSO7	51		HSI4
18		LSO6	52		HSI5
19		LSO5	53		HSI6
20		LSO4	54		HSI7
21		LSO3	55		VCC
22		LSO2	56		NC
23		LSO1	57		DI
24		LSO0	58		IT

Table 4: 8061 CPU Pinout

8061 CPU (IC-1)					
25		I/O1	59		/ST
26		I/O0	60		/PAUSE
27		EXT INT	61		MB0
28		VCC	62		MB1
29		AVSS	63		MB2
30		A10	64		MB3
31		A11	65		MB4
32		A9	66		MB5
33		A8	67		MB6
34		A12	68		MB7

Table 4: 8061 CPU Pinout

81C61/2 RAM/IO/CART (IC-3)					
1		OCP4	13		MB4
2		OCP3	14		MB3
3		OCP2	15		MB2
4		OCP1	16		MB1
5		OCP0	17		MB0
6		NC-RCV	18		NC
7		VCC	19		NC
8		NC-\XMT	20		/ST
9		ENA	21		IT
10		MB7	22		DI
11		MB6	23		NC
12		MB5	24		VSS

Table 5: 8361 RAM/IO Pinout

87C61 RAM/IO (IC-7)					
1			13	CPU-65, J3-13	MB4
2		/OE	14	CPU-64, J3-11	MB3
3			15	CPU-63, J3-9	MB2
4	GND (?)		16	CPU-62, J3-7	MB1
5			17	CPU-61, J3-5	MB0
6	GND (?)		18		
7	KAPWR	VCC	19	GND (?)	
8			20		/ST
9			21		IT
10	CPU-68, J3-19	MB7	22		DI
11	CPU-67, J3-17	MB6	23		
12	CPU-66, J3-15	MB5	24	GND	VSS

Table 6: 87C61 RAM/IO Pinout

8763 EPROM (IC-8)					
1	J3-22, 1K to +5V	/TSTSTB	13	CPU-65, J3-13	MB4
2	J3-16, 10K to +5	/EPROMDIS	14	CPU-64, J3-11	MB3
3		RP0	15	CPU-63, J3-9	MB2
4	GND	RP1	16	CPU-62, J3-7	MB1
5		RP2	17	CPU-61, J3-5	MB0
6		RP3	18	1k to +5V	VCC, VPP
7	+5	VCC	19	+5V	VCC, TEST

Table 7: 8763 EPROM Pinout

8763 EPROM (IC-8)		
8	GND	RP4
9	J3-12	/MRESET
10	CPU-68, J3-19	MB7
11	CPU-67, J3-17	MB6
12	CPU-66, J3-15	MB5
20	CPU-59, J3-21	/ST
21	CPU-58, J3-23	IT
22	CPU-57, J3-25	DI
23		/CE
24	GND	VSS

Table 7: 8763 EPROM Pinout

LEGEND:

RP	ROM PORT
OCP	OFF CHIP PORT
MB	EXTERNAL MEMORY BUS
EER	EXECUTE ENABLE REGISTER
DAR	DATA ADDRESS REGISTER
DI	DIRECTION INDICATOR
IT	INSTRUCTION TRANSFER
STB	STROBE

Table 8: 8061 Chipset Legend

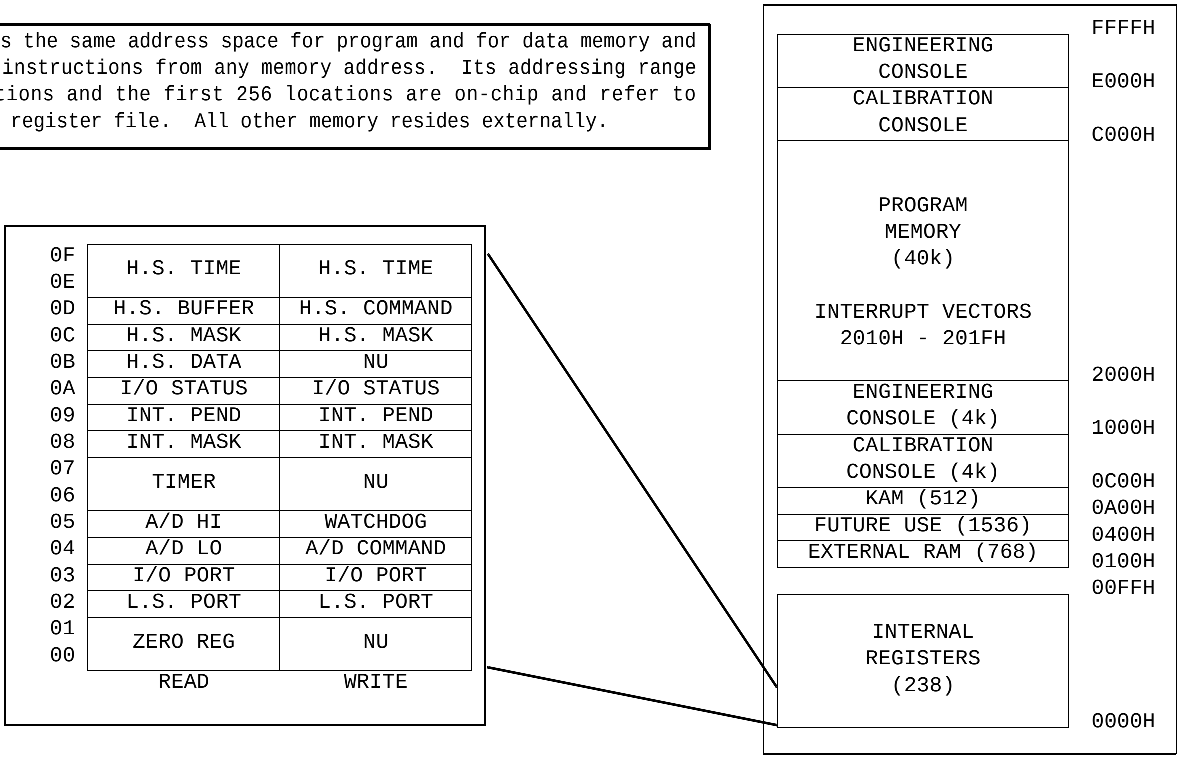
Hardware development tools used in conjunction with the EEC-IV include:

1. Engineering Console -- a lab instrument for real-time program debug and monitor of the EEC-IV system.
2. Calibration Console -- a portable unit for vehicle use to permit field display and modification of program memory.
3. D/A Converter Unit -- an add-on feature to the calibration console that converts EEC-IV system digital outputs to analog form for data logging by external recording equipment.

There is a “Production Code Release System” binary file verification and comparison program for release of production binary files to outside suppliers for ROM manufacturing.

8061 MEMORY MAP

The 8061 uses the same address space for program and for data memory and can execute instructions from any memory address. Its addressing range is 64k locations and the first 256 locations are on-chip and refer to the internal register file. All other memory resides externally.



	READ	WRITE
0F 0E	H.S. TIME	H.S. TIME
0D	H.S. BUFFER	H.S. COMMAND
0C	H.S. MASK	H.S. MASK
0B	H.S. DATA	NU
0A	I/O STATUS	I/O STATUS
09	INT. PEND	INT. PEND
08	INT. MASK	INT. MASK
07 06	TIMER	NU
05	A/D HI	WATCHDOG
04	A/D LO	A/D COMMAND
03	I/O PORT	I/O PORT
02	L.S. PORT	L.S. PORT
01 00	ZERO REG	NU

8061 INSTRUCTIONS, REGISTERS & INTERRUPTS

OP-CODE	8096	8061	8065
10	nu	nu	ROMBANK
E3	BR	nu	nu
F1	nu	RETI	RETEI
F4	nu	nu	BANK0
F5	nu	nu	BANK1
F6	nu	nu	BANK2
FD	NOP	nu	BANK3
FF	RST	NOP	NOP

Table 9: 8096-8061/5 Instruction Differences

OP-CODE	8096	8061/5	DESCRIPTION	DIFFERENCE
00	SKIP	SKP	skip - 2 byte no operation	rename
01	CLR	CLRW	clear word	rename
02	NOT	CPLW	complement word	rename
03	NEG	NEGW	negate integer	rename
04	nu			
05	DEC	DECW	decrement word	rename
06	EXT	SEXW	sign extend int to long	rename
07	INC	INCW	increment word	rename
08	SHR	SHRW	logical right shift word	rename
09	SHL	SHLW	shift word left	rename
0A	SHRA	ASRW	arithmetic right shift word	rename

Table 10: 8096-8061 Op-Codes

OP-CODE	8096	8061/5	DESCRIPTION	DIFFERENCE
0B	nu			
0C	SHRL	SHRDW	logical right shift double word	rename
0D	SHLL	SHLDW	shift double word left	rename
0E	SHRAL	ASRDW	arithmetic right shift double word	rename
0F	NORML	NORM	normalize long integer	rename
10		ROMBANK		not in 8096
11	CLRB	CLRB	clear byte	same
12	NOTB	CPLB	complement byte	rename
13	NEGB	NEGB	negate byte	same
14	nu			
15	DECB	DECB	decrement byte	same
16	EXTB	SEXB	sign extend 8-bit in to 16-bit int	rename
17	INCB	INCB	increment byte	same
18	SHRB	SHRB	logical right shift byte	same
19	SHLB	SHLB	shift byte left	same
1A	SHRAB	ASRB	arithmetic right shift byte	rename
1B	nu			
1C	nu			
1D	nu			
1E	nu			
1F	nu			
20-27	SJMP	SJMP	short jump	same
28-2F	SCALL	SCALL	short call	same
30-37	JBC	JNB	jump if bit clear	rename
38-3F	JBS	JB	jump if bit set	rename
40-43	AND	AN3W	logical and words (3 operands)	split
44-47	ADD	AD3W	add words (3 operands)	split
48-4B	SUB	SB3W	subtract words (3 operands)	split
4C-4F	MULU	ML3W	multiply unsigned words (3 operands)	split
50-53	ANDB	AN3B	logical and bytes (3 operands)	split
54-57	ADDB	AD3B	add bytes (3 operands)	split
58-5B	SUBB	SB3B	subtract bytes (3 operands)	split
5C-5F	MULUB	ML3B	multiply unsigned bytes (3 operands)	split
60-63	AND	AN2W	logical and words (2 operands)	split
64-67	ADD	AD2W	add words (2 operands)	split
68-6B	SUB	SB2W	subtract words (2 operands)	split
6C-6F	MULU	ML2W	multiply unsigned words (2 operands)	split
70-73	ANDB	AN2B	logical and bytes (2 operands)	split
74-77	ADDB	AD2B	add bytes (2 operands)	split
78-7B	SUBB	SB2B	subtract bytes (2 operands)	split
7C-7F	MULUB	ML2B	multiply unsigned bytes (2 operands)	split
80-83	OR	ORRW	logical or words	rename
84-87	XOR	XRW	logical exclusive or words	rename
88-8B	CMP	CMPW	compare words	rename
8C-8F	DIVU	DIVW	divide unsigned words	rename
90-93	ORB	ORRB	logical or bytes	rename
94-97	XORB	XRB	logical exclusive or bytes	rename
98-9B	CMPB	CMPB	compare bytes	same
9C-9F	DIVUB	DIVB	divide unsigned bytes	rename
A0-A3	LD	LDW	load word	rename
A4-A7	ADDC	ADCW	add words with carry	rename
A8-AB	SUBC	SBBW	subtract words with borrow	rename
AC-AF	LDBZE	LDZBW	load word with byte, zero extended	rename
B0-B3	LDB	LDB	load byte	same

Table 10: 8096-8061 Op-Codes

OP-CODE	8096	8061/5	DESCRIPTION	DIFFERENCE
B4-B7	ADDCB	ADCB	add bytes with carry	rename
B8-BB	SUBCB	SBBB	subtract bytes with borrow	rename
BC-BF	LDBSE	LDSBW	load integer with byte, sign extended	rename
C0/2/3	ST	STW	store word	rename
C1	nu			
C4/6/7	STB	STB	store byte	same
C5	nu			
C8-CB	PUSH	PUSHW	push word	rename
CC/E/F	POP	POPW	pop word	rename
CD	nu			
D0	JNST	JNST	jump if sticky bit is clear	same
D1	JNH	JLEU	jump if unsigned not higher	rename
D2	JGT	JGT	jump if signed greater than	same
D3	JNC	JNC	jump if carry flag is clear	same
D4	JNVT	JNVT	jump if overflow trap is clear	same
D5	JNV	JNV	jump if overflow flag is clear	same
D6	JGE	JGE	jump if signed greater than or equal	same
D7	JNE	JNE	jump if not equal	same
D8	JST	JST	jump sticky bit is set	same
D9	JH	JGTU	jump if unsigned higher	rename
DA	JLE	JLE	jump if signed less than or equal	same
DB	JC	JC	jump if carry flag is set	same
DC	JVT	JVT	jump if overflow trap is set	same
DD	JV	JV	jump if overflow flag is set	same
DE	JLT	JLT	jump if signed less than	same
DF	JE	JE	jump if equal	same
E0	DJNZ	DJNZ	decrement and jump if not zero	same
E1	nu			
E2	nu			
E3	BR		branch indirect	not in 8061/5
E4	nu			
E5	nu			
E6	nu			
E7	LJMP	JUMP	long jump	rename
E8	nu			
E9	nu			
EA	nu			
EB	nu			
EC	nu			
ED	nu			
EE	nu			
EF	LCALL	CALL	long call	rename
F0	RET	RET	return from subroutine	same
F1	-----	RETEI/RETI	retei-8065 reti - 8061	not in 8096
F2	PUSHF	PUSHP	push flags	rename
F3	POPF	POPP	pop flags	rename
F4		BANK 0	8065 only - not in 8096	
F5		BANK 1	8065 only - not in 8096	
F6		BANK 2	8065 only - not in 8096	
F7	TRAP	INT	software trap (internal use only, not in assembler)	
F8	CLRC	CLC	clear carry flag	same
F9	SETC	STC	set carry flag	rename
FA	DI	DI	disable interrupt	same
FB	EI	EI	enable interrupt	same

Table 10: 8096-8061 Op-Codes

OP-CODE	8096	8061/5	DESCRIPTION	DIFFERENCE
FC	CLRVT	CLRVT	clear overflow trap	same
FD	NOP	BANK 3	NOP 8096/8061	different
FE	PREFIX	SIGND/ALT	changes multiply/divide to signed	rename
FF	RST	NOP	system reset in 8096 NOP in 8061/5	different

Table 10: 8096-8061 Op-Codes

The bank selection opcodes are 8065 -- as that is the difference between them, memory bank selection capabilities.

	00	10	20	30	40	50	60	70
00	SKP	ROMBANK*	SJMP	JNB0	AN3W d	AN3B d	AN2W d	AN2B d
01	CLRW	CLRB	SJMP	JNB1	AN3W =	ANDB =	AN2W =	AN2B =
02	CPLW	CPLB	SJMP	JNB2	AN3W @	AN3B @	AN2W @	AN2B @
03	NEGW	NEGB	SJMP	JNB3	ANEW ()	AN3B ()	AN2W ()	AN2B ()
04	(ua)	(ua)	SJMP	JNB4	AD3W d	AD3B d	AD2W d	AD2B d
05	DECW	DECB	SJMP	JNB5	AD3W =	AD3B =	AD2W =	AD2B =
06	SEXW	SEXB	SJMP	JNB6	AD3W @	AD3B @	AD2W @	AD2B @
07	INCW	INCB	SJMP	JNB7	AD3W ()	AD3B ()	AD2W ()	AD2B ()
08	SHRW	SHRB	SCALL	JB0	SBEW d	SB3B d	SB2W d	SB2B d
09	SHLW	SHLB	SCALL	JB1	SB3W =	SB3B =	SB2W =	SB2B =
0A	ASRW	ASRB	SCALL	JB2	SB3W @	SB3B @	SB2W @	SB2B @
0B	(ua)	(ua)	SCALL	JB3	SB3W ()	SB3B ()	SB2W ()	SB2B ()
0C	SHRDW	(ua)	SCALL	JB4	ML3W d	ML3B d	ML2W d	ML2B d
0D	SHLDW	(ua)	SCALL	JB5	ML3W =	ML3B =	ML2W =	ML2B =
0E	ASRDW	(ua)	SCALL	JB6	ML3W @	ML3B @	ML2W @	ML2B @
0F	NORM	(ua)	SCALL	JB7	ML3W ()	ML3B ()	ML2W ()	ML2B ()

Table 11: Op-Code Map 00-70

	80	90	A0	B0	C0	D0	E0	F0
00	ORRW d	ORRB d	LDW d	LDB d	STW d	JNST	DJNZ	RET
01	ORRW =	ORRB =	LDW =	LDB =	(ua)	JLEU	(ua)	RETI/RETEI*
02	ORRW @	ORRB @	LDW @	LDB @	STW @	JGT	(ua)	PUSHP
03	ORRW ()	ORRB ()	LDW ()	LDB ()	STW ()	JNC	(ua)	POPP
04	XRW d	XRB d	ADCW d	ADCB d	STB d	JNVT	(ua)	BANK0*
05	XRW =	XRB =	ADCW =	ADCB =	(ua)	JNV	(ua)	BANK1*
06	XRW @	XRB @	ADCW @	ADCB @	STB @	JGE	(ua)	BANK2*
07	XRW ()	XRB ()	ADCW ()	ADCB ()	STB ()	JNE	JUMP	INT**
08	CMPW d	CMPB d	SBBW d	SBBB d	PUSHW d	JST	(ua)	CLC
09	CMPW =	CMPB =	SBBW =	SBBB =	PUSHW =	JGTU	(ua)	STC
0A	CMPW @	CMPB @	SBBW @	SBBB @	PUSHW @	JLE	(ua)	DI
0B	CMPW ()	CMPB ()	SBBW ()	SBBB ()	PUSHW ()	JLC	(ua)	EI
0C	DIVW d	DIVB d	LDZBW d	LDSBW d	POPW d	JVT	(ua)	CLRVT
0D	DIVW =	DIVB =	LDZBW =	LDSBW =	(ua)	JV	(ua)	BANK3*
0E	DIVW @	DIVB @	LDZBW @	LDSBW @	POPW @	JLT	(ua)	ALT*/SIGND/S
0F	DIVW ()	DIVB ()	LDZBW ()	LDSBW ()	POPW ()	JE	(ua)	NOP

Table 12: Op-Code Map 80-FF

Priority:	Interrupt	16-Bit Address
Highest	High-Speed Input #0	0x201E

Table 13: 8061 Interrupt Vectors and Priorities

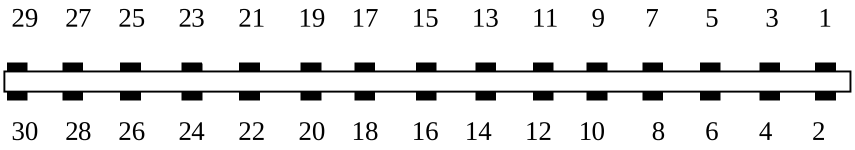
Priority:	Interrupt	16-Bit Address
High	High-Speed Input #1	0x201C
High	HSO Port Output Interrupt #1	0x201A
Low	External Interrupt	0x2018
Low	HSI Port Input Data Available	0x2016
Low	A/D End-Of-Conversion	0x2014
Low	Master I/O Timer Overflow	0x2012
Lowest	HSO Port Output Interrupt #2	0x2010

Table 13: 8061 Interrupt Vectors and Priorities

At Reset, PC = 0x2000 in Memory Bank #8

addr	READ	WRITE
0000:0001	ZERO	R/O
0002	LOW SPEED I/O uni-directional	
0003	I/O bi-directional	
0004:	A/D LO	A/D channel #
0005:	A/D HI	watchdog timer
0006:0007	I/O timer	R/O
0008:	INTERRUPT MASK	INTERRUPT MASK
	bit 7: HSI0 bit 6: HSI1 bit 5: HSO output int 1 bit 4: External interrupt bit 3: HSI data ready bit 2: A-D complete bit 1: I/O timer ovflw bit 0: HSO int 2	
0009:	INTERRUPT STATUS (same mask as above)	
000A:	IO STATUS	IO STATUS
	bit 5: interrupt priority 1/0 hi/low	
	bit 4: interrupt status 1/0 int serv/prog exe	
	bit 3: A-D complete	
	bit 2: HSI data ready	
	bit 1: HSI input ovflw	
	bit 0: HSO output ovflw	
000B:	HSI sample	R/O
000C:	HSI input mask	HSI input mask
000D:	HSI data hold	HSO command hold (write byte after HSO time) bits 0-3: select HSO channel bit 4: output level bit 5: interrupt enable 0FH = software interrupt
000E:000F	HSI time hold	HSO time hold (write word before HSO command)
0010:0011	STACK POINTER	STACK POINTER
0012:00FF	general register file	general register file

Table 14: Register Map



(View facing connector on PCM.)

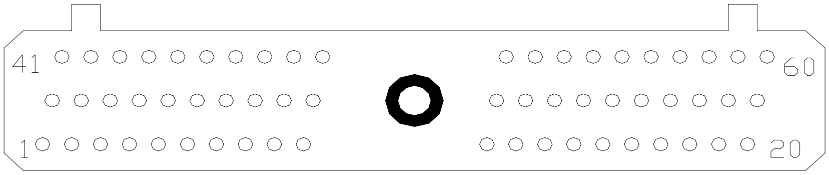
ECM TEST PORT (J3)

The pinouts are derived from the J3 Test Port on a SD unit for an '87 Mustang (DA1 / E7SF-12A650-A1B). Looking at the MCU facing the service port (from the rear of the mating plug) the connector is numbered from right-to-left with odd numbers on the component side and the even numbers on the wiring side. It is a 15/30 terminal, card-edge connector with .1" spacing. (The table below is arranged for the pins to be read from left-to-right, top first.)

PIN	SIGNAL / FUNCTION	MCU PIN	CPU 8061	RAM 81C61	EPROM 8763	notes
29	PWR GND	40,60				
27	VPWR	37,57				battery +
25	DI		57	22	22	
23	IT		58	21	21	
21	STROBE\		59	20	20	
19	D7		68	10	10	
17	D6		67	11	11	
15	D5		66	12	12	
13	D4		65	13	13	
11	D3		64	14	14	
9	D2		63	15	15	
7	D1		62	16	16	
5	D0		61	17	17	
3	KA5V			7		
1	VREF (+5)	26				
30	PWR GND					
28	VPWR	37,57				battery +
26	BS0					
24	BS3					
22	TSTSTB\				1	1K TO +5 only
20	PROGRAM					
18	RAMDISABLE\					
16	EPROMDISABLE\				2	10K TO +5 only
14	ERASE					
12	MRESET\				9	
10	RESET\		3	9		
8	PAUSE\		60			1K TO +5 only
6	EXTINT					
4	(high for access)					IC4-74001 pin 13
2	VCC	25				

Table 15: ECM Test Port (J3) Pinout

There're 14 pins from the 8763 EPROM on the connector, 2 pins from the 87C61 RAM-I/O on the connector, 1 pin from the 8061 CPU and 1 pin from a 16-pin logic chip.



(looking at connector from outside MCU)

ECM WIRING HARNESS

The table below lists several MCU cabling pinouts. The first is for a Mustang A9L EEC, used 1989 to 1993 302 MAF, manual transmission applications. The second, for a 1991 Ranger 2.3L Dual Plug EFI Engine (from Mitchell International On-line manual (c) 1992), was submitted by <tnye@mansci.watstar.uwaterloo.ca>. The next three columns were done by the author and are from the Ford wiring diagrams (yes, I actually buy the factory manuals) -- and they don't use the same naming convention as the first two.

No	Mustang A9L	'91 Ranger 2.3	'91 4.9, 5.0, 5.8 F-series w/o E4OD	'91 4.9, 5.0, 5.8 F-series w E4OD	'91 7.3 diesel F-series w E4OD	Description
1	Kapwr	Kapwr	Bty to load	Bty to load	Bty to load	keep-alive power
2		BOO		stop lamp sw to turn sig sw	stop lamp sw to stop lamp	Brake On/Off switch
3	VSS +	VSS +	VSS	VSS	VSS	Vehicle speed sensor positive
4	IDM	IDM	elect sw to ignition coil	elect sw to ignition coil	elect sw to ignition coil	Ignition Diagnostic monitor
6	VSS -	VSS -	VSS return	VSS return	VSS return	Vehicle speed sensor negative
7	ECT	ECT	engine coolant	engine coolant	xmsn oil temp	Engine coolant temp sensor
8	FPM	FPM	fuel sply pump relay	fuel sply pump relay		Fuel pump monitor
9		DATA -	EEC data -	EEC data -		
10	ACC	ACC	compressor clutch feed	compressor clutch feed		A/C compressor clutch
11	AM 2		thermactor diverter valve	thermactor diverter valve		Air management solenoid 2
12				4X4 lo-range indicator	4X4 lo-range indicator	
14		MAF (CA only)				
15		MAF RTN (CA only)				
16	IGN GND	IGN GND	dedicated GND to TFI	dedicated GND to TFI		Ignition ground
17	STO/MIL	STO/MIL	check engine lamp	check engine lamp	check engine lamp	Self-test output check Engine
20	CSE GND	CSE GND	GND	GND	GND	Case ground
21	ISC/BPA	ISC/BPA	ISC	ISC		Idle speed control / bypass air

Table 16: ECM CABLE PINOUT

No	Mustang A9L	'91 Ranger 2.3	'91 4.9, 5.0, 5.8 F-series w/o E4OD	'91 4.9, 5.0, 5.8 F-series w E4OD	'91 7.3 diesel F-series w E4OD	Description
22	FP	FP	fuel pump relay	fuel pump relay		Fuel pump
23	KS		KS	KS		Knock sensor
24	PSPS	PSPS	PSPS	PSPS		Power steering pressure switch
25	ACT	ACT	ACT	ACT		Air charge temperature
26	VREF	VREF	pwr to sensors	pwr to sensors	pwr to sensors	Reference voltage
27	EVP	HEGO	EGR posn feed	EGR posn feed		EGR valve position sensor
28		NDS	EEC data +	EEC data +		
29	HEGO	HEGO	EGO	EGO		Heated exhaust gas oxygen sensor
30	NDS	NDS/CES	fuel sensor GND	fuel sensor GND	man'l lever pos or clutch	Neutral drive switch (automatic)
31			canister purge solenoid	canister purge solenoid		
32				O/D cancel lamp	O/D cancel lamp	
33	EVR	EVR	EGR valve	EGR valve		EGR vacuum regulator solenoid
34						DOL
35					EEC to xmsn	
36	SPOUT	SPOUT	SPOUT from TFI	SPOUT from TFI	tach feed	Spark out timing control
37	VPWR	VPWR	pwr rly to EEC	pwr rly to EEC	pwr rly to EEC	Vehicle power
38				electronic pressure ctl	electronic pressure ctl	
40	PWR GND	PWR GND	dedicated GND	dedicated GND	dedicated GND	Power ground
41				xmsn O/D switch	xmsn O/D switch	
42				xmsn oil temp		
43		ACD	A/C demand	A/C demand		
44					tach retn	
45	MAP	MAP	MAP feed	MAP feed	MAP feed	Manifold absolute pressure
46	SIG RTN	SIG RTN	sensor sig rtn	sensor sig rtn	sensor sig rtn	Signal return
47	TPS	TPS	TAPS	TAPS	TAPS	Throttle angle position sensor
48	STI	STI	EEC to test conn #1	EEC to test conn #1	EEC to test conn #1	Self-test input
49	HEGOG	HEGOG	fuel sensor GND	fuel sensor GND		Heated EGO sensor ground
51	AM 1		thermactor dump valve	thermactor dump valve		Air management solenoid 1
52		SS		xmsn throttle valve sol #1	xmsn throttle valve sol #1	
53		CCO		clutch sw	clutch sw	Converter Clutch Overide
54		WAC				
55				coast clutch sol	coast clutch sol	
56	PIP	PIP	PIP from TFI	PIP from TFI		Profile ignition pickup
57	VPWR	VPWR	pwr rly to EEC	pwr rly to EEC	pwr rly to EEC	Vehicle power
58	INJ 1	INJ 1	INJ 1	INJ 1		Injector bank 1
59	INJ 2	INJ 2	INJ 2	INJ 2		Injector bank 2
60	PWR/GND	PWR/GND	dedicated GND	dedicated GND	dedicated GND	Power ground

Table 16: ECM CABLE PINOUT

Technical Notes on The EEC-IV MCU

SOFTWARE

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

THE BASICS

The EEC-IV original objectives included "calculating all required engineering command values and output the required real time commands for spark timing, exhaust gas recirculation valve positioning and for turning fuel injectors on or off in a sequentially firing configuration, in a total time of 2.5 milliseconds or less."

EEC-IV programs are 57k in length and EEC-V programs are 114k and they both begin with:

```
   FF       NOP
   FA       DI
```

For EEC-IV there are 2 types V2 (64k) and V5 (32k). The EEC-IV V5 is the only case where all the software and data is in a single chip (32k). Early Ford models (1990-1993). For EEC-V (114k source file) the board uses 57k only (the maps are in the 2nd part of the source) and the processor inside the board makes the right addressing function.

The EEC-IV tables can be located by looking at the cal pointers (located somewhere in the 30 or 50 bytes of EPROM after 2000). The table ’segment’ addresses are in there. Another helpful hint is that Ford likes the binary point method of scaling.

INTERNAL DIAGNOSTICS

Two types of diagnostics are performed by the EEC (this was written for early 80’s model units so it may be expanded now). They are On-Demand and Continuous diagnostics. On-Demand is conducted during key-on/engine-off and during engine running modes to permit the microprocessor to test itself. Continuous, as the name implies, is on-going whenever the system is in operation. Beginning in the latter part of 1983, the EEC-IV began to remember conditions found during continuous testing, even after the key is turned off with a special custom memory chip called Keep Alive Memory (KAM). The KAM chip, which contains 128 bytes of read/write memory, is powered by a separate low current connection to the vehicle battery. Faults, even intermittent ones, are recognized and stored away for recall during dealer service.

FUEL CONTROL

There are two basic methods of fuel delivery: throttle body injection and manifold injection, and the methods of determining how much fuel to be delivered is determined using speed density, air volume or mass air techniques. The volume or mass air approaches require sensors to determine those values. The Air Flow sensor used in production EFI’s typically compensates for temperature and density changes in the intake air mass, then the oxygen sensor is used to fine tune the mixture. Almost all use barometric compensation in one form or another. Some systems take a barometric reading from the MAP sensor after the ignition key is turned on, but before the engine starts, and store this as a reference. This can also be updated at WOT, since manifold pressure is essentially = barometric pressure at this point (with some flow related pressure drop). Some systems have a separate barometric sensor in addition to MAP. Some MAP's are not absolute sensors at all, but differential sensors, referenced on one side to the atmosphere. So as the atmospheric pressure changes, the MAP reference point changes as well. Some compensation is possible with the fuel pressure regulator, since it is usually referenced to manifold pressure and thus atmospheric indirectly. This helps regulate the pressure across the injector so the amount of fuel delivered is related to only the injector pulse width. Some systems have no barometric pressure compensation at all.

The EEC does 4 point interpolation on all tables. There is a minimal number of cells in the fuel lookup tables. The EEC doesn't look up 'injector on time', it calculates the injector pulse width by looking at the desired Lambda and then, using the mass of air entering the engine and the injector size, it calculates the duty cycle needed to get the desired A/F ratio. (Lambda is an engineering term where stoich is 1, anything smaller than 1 is rich, anything larger than 1 is lean. To get A/F numbers from Lambda, multiply lambda value by 14.64. For example, an A/F ratio of 14.05:1 is a lambda of .85 lambda.)

The ECU controls both the fuel mixture and the timing with the fuel mixture operating in either "open loop" or "closed loop" mode. Anything external to the EEC that tries modify the fuel mixture at points where the engine is in closed loop operation will cause the computer to attempt to compensate. That's why trying to change the A:F ratio and "fool the computer" cause more problems than it's likely to solve. Timing and WOT fuel settings aren't closed loop functions, and can be changed without the computer trying to correct them. This is why units that connect between the cable and the ECU aren't very effective at modifying fuel mixtures.

Closed loop operation can sometimes be altered without problems. This has allowed some manufacturers to market cars and parts that are fully emissions legal (e.g. Kenne-Belle, Saleen, etc). After-market devices that go between the engine harness and the EEC interfere with closed loop. Software modules that connect to the service connector (Hypertech, Superchips, etc.) do not interfere with closed loop - rather they can define new values for closed loop. The EEC will do whatever it's told -- it's a computer running a program and your data can be substituted for the factory's through the service port connector. The EEC cannot 'learn' around a software module.

Closed loop operation basically consists of a controller with a target A/F ratio, HEGO information as its feedback and the injectors as the main control mechanism. The 'factory' target A/F ratio is 14.64:1, but this can be changed.

Approximately 900 items can be changed or logged in a 93 5.0 Mustang. For example, during a shift, the EEC might look at spark, load, TP, fuel, and transient fuel. By logging this data, you can tell exactly where in the spark tables the EEC is travelling and tune just those cells. Most people would normally tweak the whole curve down or try and tune in areas the EEC isn't even looking at. With data-logging, you can see exactly where it's pulling its data from.

Examples of some of the functions controlled by the EEC are: A:F ratio in closed loop, transient fuel, EGR, Canister Purge, Thermactor, adaptive control system, control of OBD-I and OBD-II testing (on/off/change test values...), fuel, spark, MAF's, VE tables, injectors, rev limits, speed limits, electronic transmission control, and lots more.

If you have a later car (91 or newer), there is an integrated controller module (ICM) (12B577 basic #) located in the engine compartment. It is a black metal box about 8"X6"X1.5" and runs the cooling fan, the fuel pump, and the EEC power.

A contributor wrote: "I've been hacking away at the Taurus code, and I think I have most of the code broken down into chunks. It looks as though Ford calculates AIRFLOW from the speed-density calculation, then re-calculates AIR MASS PER CHARGE from it. I’m having a hard time figuring out the spark timing algorithms, probably half way through the PFE EGR system, and I’ve looked at the O2 system. The adaptive learn stuff is weird. I’ve looked briefly at the self-test. The timer scheduler is very weird. They ran out of HSO slots by the looks of things, so they wrote some clever code to get around that."

IGNITION AND TIMING CONTROL:

The EEC only sees one Crankshaft Position Sensor signal, but where it comes from depends on the age of the EEC. Early EEC’s used a sectored wheel in the distributor which produced a square wave of frequency of Number-Cylinders per 2-revs with a nominal 50% duty cycle unless SEFI was used whereupon there was a "short" tooth. The spark was output by a TFI unit.

Later and perhaps all current EEC’s, including the EEC-V, utilize a 36-1 tooth wheel for CPS which is pre-processed by a unit known as the EDIS (Electronic DIStributor). The EDIS converts the 36-1 into a 2 pulses/rev 50% duty cycle square wave which is then fed into the EEC to be used for RPM and injector timing calculations. The EEC sends a PWM signal to the EDIS defining the spark advance required, and the EDIS unit then times out the signals to the coils (wasted spark). This gives a more accurate spark delivery as the EDIS has access to timing data which is updated every 10 crank degrees whereas the EEC only gets timing data every 90 degrees.

The EEC gets one and only one timing signal from the TFI unit. It is called the PIP (Profile Ignition Pickup). The PIP signal is 45 - 55Hz @ 1000 RPM, for 4, 6 and 8 cylinder engines and, with the exception of SEFI, has a duty cycle of 50%. SEFI uses Signature PIP where the #1 vane on the PIP reluctor is roughly 35% duty cycle and the rest are roughly 50%. The EEC uses this to detect cylinder #1. On a stock car, the leading edge of the PIP signal is @ 10 BTDC.

The EEC controls the spark timing. The TFI’s function at this point is to basically clean up the PIP signal and charge and fire the coil. The TFI module conditions the hall sensor output and sends it off to the EEC. The only delay is just propagation delay through the TFI electronics. The EEC sends out the SPOUT signal which starts the TFI modules charging the coil. Depending on what advance the EEC is looking for, the falling edge of the SPOUT can vary. The coil fires on the falling edge. Since the EEC ’knows’ where 10 BTDC of each cylinder is, it can calculate when to drop the SPOUT signal by using timers. The MCU uses the previous PIP value to determine where the crank was. The TFI module can handle acceleration rates of up to 250 HZ/sec. Another function of the TFI modules is to provide LOS spark (limp mode). If the TFI detects a loss of SPOUT, it will generate it’s own ’SPOUT’ to coincide with the rising edge of PIP (10 BTDC...assuming you haven’t moved the distributor).

To determine timing values, the EEC uses crank position (CPS), engine temperature (ECT), air-charge temperature (ACT), throttle position (TPS), EGO data and Cylinder-ID to name the significant ones. It’s relatively easy to calculate the spark required for optimum power from these, but the compromises made to meet emissions and driveability complicate matters.

The "TFI" (EDIS) units are all very similar. The differences are in the EECs which, though electrically similar, are totally different in terms of code and calibration content. The EDIS gets the required spark advance from the EEC and, using the regularly updated crankshaft position, determines the ignition firing time.

The return from the EEC to the TFI module (SPOUT or SPark OUT) is the timing information and has the same specifications as PIP. What was gleaned from this is that the PIP does 2 things:

1) It lets the EEC know how fast the engine is turning (frequency alone).

2) It gives a base signal to be sent back to the TFI after being delayed a bit. This delay or phase change (relative to the PIP) is what lets the EEC control timing. But, indirectly, the TFI is doing most of the work.

The return signal from the EEC to the EDIS is unrelated to the PIP. It purely indicates to the EDIS unit the amount of spark advance required.

Knock Sensor

It looks like the SES/STO line is used to set the knock sensor threshold by pulsing the line to control the knock threshold. This only applies to systems using knock sensors. Does anyone know of any other rogue use of the STO line? If yes, does anyone have another suggestion on a line to use to sneak data out on? I am assuming that the EEC everyone is concentrating on does not have a CART like the Taurus one does.

LIMITED OUTPUT STRATEGY (LOS) OPERATION

The CPU’s watchdog timer provides pulses on pin 18 (J2-15) which periodically retrigger multivibrator IC7 (74001MC) on the daughter PCB. If the CPU does not retrigger the multivibrator within a time interval specified by the time constant of C18 & R52, then IC7-7 goes high, thereby enabling LOS operation.

PIP inputs then trigger the second half of IC7 and produce injector output pulses at IC7-10. The width of these pulses is defined by the time constant of R46 & C17.

IC7-7 enables the preset LOS inputs of IC4 & IC8, and selects the fixed width injector pulses produced by IC7. IC1-2 (70002BB) disables IC3 & IC6 (71001FB). The resulting outputs are as follows:

Self test output	ON
Injector banks 1 & 2	fixed pulse width, triggered by PIP
Fuel pump relay	ON
pin # 52, Q17	OFF
EGR vacuum regulator solenoid	OFF
pin # 53, not populated	OFF
Idle speed control solenoid	OFF
Air management #1	OFF
Air management #2	OFF
pins 31,32,34,35,54,55	OFF

All inputs are ignored.

IC1 senses VPWR and enables LOS operation if VPWR is below acceptable level.

FUNCTIONS / SCALARS / TABLES

These are the known parameters used in the EEC’s engine management algorithms and the search is continuing for others. Not all of these are used in every EEC application and there are likely to be others that have yet to be found. The actual memory locations, where known, are listed for the A9L.

Dependent Variable	Independent Variable	Function	A9L
EEC Load Scaling	Raw Load	Table Row Pointer	6CB2
MAF Transfer	MAF Voltage	Mass Flow Kg/hr	6DE2
WOT Spark Advance	RPM	WOT Spark Advance Deg BTDC	7504
WOT Spark Advance	ECT	WOT Spark Advance Deg BTDC	7532
WOT Spark Advance	ACT	WOT Spark Advance Deg BTDC	7540
Accelerator Enrichment Multiplier	TP - A/D Counts	Accelerator Enrichment Multiplier	71C1
WOT Fuel Multiplier	RPM	WOT Fuel Multiplier	7B80
Open Loop Fuel Multiplier	ACT	Open Loop Fuel Multiplier	7B48
Spark Advance	ACT	Spark Advance [Deg BTDC]	74EE
Spark Advance	Barometric Pressure	Spark Adder Deg	7528
Spark Advance Rate	RPM	Seconds/Degree	779C
Min Low Speed Dwell	Battery Voltage	Dwell / Coil Off Time [ms]	7748
Min High Speed Dwell	Battery Voltage	Dwell / Coil Off Time [ms]	7760
Maximum Dashpot Clip	RPM	Dashpot Clip [lb/min]	7346
Sea Level Lugging O/L Fuel Multiplier	RPM	Lugging O/L Fuel Multiplier	782A
Altitude Lugging O/L Fuel Multiplier	RPM	Lugging O/L Fuel Multiplier	7846
Crank Fuel Pulse Width Multiplier	Time in Crank Mode	Crank Fuel Multiplier	729A
Cranking Fuel Pulsewidth	ECT	Pulsewidth [ms]	787A
Injector Offset vs. Battery Voltage	Battery Voltage	Injector Offset	7B30
Open Loop Fuel Multiplier	RPM	Open Loop Fuel Multiplier	7DE8
Dashpot Decrement Rate	Dashpot[lb/min]	Decrement Rate [lb/min]	732A
WOT Fuel Multiplier	TP		
Part Throttle Spark Advance	ACT		
Closed Throttle Open Loop Fuel Multiplier			
Altitude Fuel Adjustment			
Cranking Fuel	ECT		
Transmission TV Pressure	TP		
Torque Converter Lockup	TP		
Upshift Speed	TP		
Downshift Speed	TP		
Idle Airflow			

Table 17: EEC Functions

EEC SCALARS

Scalar	A9L
Injector Size	
Injector Slope	
Minimum Injector Pulse Width	
Accelerator Pump Multiplier	
Open Loop Fuel Multiplier	
Part Throttle Timing Adder	
Dwell Minimum	
Dwell Maximum	
ACT Minimum for Adaptive Control	
ACT Maximum for Adaptive Control	
Minimum ECT for Deceleration Fuel Shutoff	
Minimum RPM for Deceleration Fuel Shutoff	
Minimum Load (MAP) for Closed Loop	
Hi-load Timeout to Open Loop	

Table 18: EEC Scalars

Scalar	A9L
Idle Speed Neutral	
Idle Speed Drive	
CID	
Number HEGO Sensors	
WOT TPS value	
EGR Multiplier	
EGR Type	
PIP Filter	* 7C16
Half Fuel Rev Limit ON	7A82
Half Fuel Rev Limit OFF	7A84
Speed Limit A -- ON	* 7A86
Speed Limit A -- OFF	* 7A87
Speed Limit B-- ON	* 7A88
Speed Limit B-- OFF	* 7A89
Maximum Spark Retard	
Cooling Fan ECT HI/LO/Hysteresis	
Intake Manifold Volume	
Thermactor Presence	

Table 18: EEC Scalars

Scalar Examples

Scalars for ‘86-’93 Mustang from Calibrator Demo	
Injector Size (lbs/hr)	19.64
Injector Slope	0.0072494
Global Accel Pump Multiplier	0.40
Gobal Open Loop Fuel Multiplier	1.00
Global Injector Delay (Crank Deg)	0
Closed Throttle Spark Adder (Deg BTDC)	0
Part Throttle Spark Adder (Deg BTDC)	0
WOT Spark Adder (DEG BTDC)	0
Tip-in Spark Retard (Deg BTDC)	0
Maximum Lo-Speed Dwell (OFF Time in mS)	0.50000
Maximum Hi-Speed Dwell (OFF Time in mS)	0.20000
ACT Minimum for Adaptive Control (Deg F)	100
ACT Maximum for Adaptive Control (Deg F)	200
Minimum ECT for Deceleration Fuel Shutoff (Deg F)	140
Minimum RPM for Deceleration Fuel Shutoff	149.75
Minimum Load (MAP) for Closed Loop	.0050049
Hi-load Timeout to Open Loop (Seconds)	2
Idle RPM in Neutral	672
Idle RPM in Drive	672
CID	301.04
Number HEGO Sensors	2
WOT TPS Breakpoint (A/D Counts)	555
EGR Multiplier	1
PIP Filter (CPU Cycles/Crank Rev)	961
Turn ON Half Fuel Rev Limit	6100
Turn OFF Half Fuel Rev Limit	6100
Maximum RPM (No Fuel)	6250
Turn ON 1st Speed Limiter (MPH)	127.5
Turn OFF 1st Speed Limiter (MPH)	127.5
Turn ON 2nd Speed Limiter (MPH)	127.5
Turn OFF 2nd Speed Limiter (MPH)	127.5

‘94-5 Mustang Scalars from Calibrator Demo	
High Injector Slope (lb/hr)	19.641
Low Injector Slope (lb/hr)	28.8
Injector Breakpoint	1.0014e-005
Min Injector Pulsewidth	0.0014
ACT Minimum for Adaptive Control (Deg F)	-20
ACT Maximum for Adaptive Control (Deg F)	160
Global Spark Adder	0
Max Degrees of Spark Retard	-15
Minimum Tip-in Spark (Deg)	5
Gobal Open Loop Fuel Multiplier	1.00
WOT TP Breakpoint (V)	3.0518
ECT to Turn ON HS Cooling fan	228
ECT to Turn OFF HS Cooling fan	220
ECT to Turn OFF LS Cooling fan	204
ECT to Turn ON LS Cooling fan	208
Min ECT to Allow Decel Fuel Cut-off	140
Intake Manifold Volume	4
Open Loop Idle Multiplier	1
CID	301.04
EGR Multiplier	1
EGR Type (2=None)	0
Thermactor Present (0/1 = NO/YES)	1
Enable Stage 1 Speed Limiter	10000
Disable Stage 1 Speed Limiter	10000
Enable Stage 2 Speed Limiter	10000

Table 19: Example Scalars from Calibrator Demo

EEC TABLES

Parameter	Variable	Variable	A9L
Accel Enrichment Fuel [lb/min]	Throttle Rate [Deg/sec]	ECT	01CB
Startup Fuel [A/F Ratio]	ECT	Time	7C40
Base Fuel [A/F Ratio]	ECT	Load	7BF0
Injector Timing [Crank Degrees]	RPM	Load	7160
Load Injector Firing Order	Injector to Fire		71B9
Base Spark [Deg BTDC]	RPM	Load	7574
Altitude Base Spark [Deg BTDC]	RPM	Load	75C4
Limp Mode Spark Table [Deg BTDC]	RPM	Load	7614
Injector Output Port	Injector Port		71B1
Borderline Detonation Spark			
Borderline Compensation	ECT	ECT	
Borderline Compensation	ACT	ACT	
Borderline Compensation	Lambda	Lambda	
Acceleration Fuel Time Constant			
Exhaust Pulse Delay			
HEGO Amplitude			
HEGO Bias			
Engine Torque			
Engine Frictional Torque			

Table 20: EEC Tables

A9L Constants & Locations

Below is a description of each parameter stored in the EEC computer -- so far as we know it. For a more detailed description of how the data is actually stored in the EEC, go to the section titled “A9L Tables” on page 39 and “A9L Table Format Explanation” on page 43.

		THROTTLE RATE DEG/SEC							
		15	33	48	63	77	90	110	125
ECT	200	1.5	1.5	2.0	3.0	4.5	8.0	12.0	18.0
	150	1.5	1.5	2.0	3.0	4.5	8.0	12.0	20.0
	100	2.0	2.0	3.0	4.5	8.0	12.5	18.0	29.0
	50	4.0	5.0	6.0	8.0	12.5	17.0	25.5	34.0
	30	5.0	8.0	10.0	12.0	16.5	21.0	27.0	45.0
	0	12.0	14.0	16.0	18.0	26.0	30.0	38.0	55.0
	-30	21.0	23.0	26.0	28.0	31.0	34.0	39.0	56.0

Table 21: Accel Enrichment Fuel [lb/min]

The **Accelerator Enrichment Fuel Table** provides an operation similar to the "accelerator pump" in a carburetor. It gives amounts of extra fuel to add based on how fast the throttle is being depressed and modifies that by the engine temperature. For the A9L, the table is located at 71CB hex, is 8 x 7 in size, the values being stored as unsigned, single byte values in a two binary point representation (i.e. the smallest resolution is .25 lb/min). To get the values below, the value of each corresponding byte will need to be divided by four or multiplied by .25.

ENGINE COOLANT TEMPERATURE ^{O}F									
-30	0	30	60	90	120	150	180	210	240

Table 22: Startup Fuel Ratio

	45.00	0	0	0	0	0	0	0	0	0	0
	35.00	1.000	1.25	0.875	0.500	0.375	0.875	0.875	0.250	0.500	1.000
T	**25.00**	1.250	1.500	1.125	1.000	1.000	1.250	1.250	0.875	1.000	1.250
I	**15.00**	1.625	1.625	1.500	1.375	1.250	1.375	1.375	1.125	1.375	1.625
M	**10.00**	2.250	2.125	2.125	1.875	1.875	1.875	1.875	1.250	1.875	2.250
E	**5.00**	2.750	2.750	2.375	1.875	1.875	1.875	1.875	1.500	1.875	2.750
	3.00	2.875	3.625	2.625	1.875	1.875	1.875	1.875	1.625	1.875	2.875
	0.00	3.375	4.125	3.000	1.875	1.875	1.875	1.875	1.375	1.875	3.375

Table 22: Startup Fuel Ratio

The **Startup Fuel A:F Ratio Table** sets the startup fuel mixture enrichment based on the engine temperature -- analogous to the choke function of a carburetor. The mixture is additionally modified by the time the engine has been operating. For the A9L, this data table is stored at 7C40. It is a 10 x 8 table with unsigned, single byte values stored in three binary point format (.125 resolution). To retrieve the information from the byte values, they must be divided by 8.

ENGINE COOLANT TEMPERATURE ^{O}F

		-30	**0**	**30**	**60**	**90**	**120**	**150**	**180**	**210**	
	90	10.500	10.500	12.500	12.500	12.750	13.125	13.500	13.500	13.500	13.500
L	**80**	11.250	11.250	12.7500	12.750	13.125	13.500	13.750	13.750	13.625	13.500
O	**70**	12.125	12.125	13.500	13.500	13.875	14.375	14.125	14.250	13.625	13.625
A	**55**	12.750	13.875	14.250	14.375	14.875	14.875	15.250	16.000	16.000	15.000
D	**40**	13.625	14.375	14.375	14.375	14.875	15.250	15.250	16.000	16.000	14.750
	30	14.125	14.250	14.250	14.375	14.500	14.750	14.750	15.250	15.250	14.375
%	**15**	17.625	17.625	17.000	16.625	16.500	16.375	16.000	15.250	15.250	15.250
	5	17.875	17.875	17.625	17.250	16.625	16.375	16.000	15.250	15.250	15.250

Table 23: Base Fuel A:F Ratio

Base Fuel A:F Ratio: ECT vs. Load. Located at 76F0 (for the A9L), it is a 10 x 8 table of single, unsigned binary values stored in three binary point format (divide by 8).

RPM

L	420	420	420	420	420	420	420	464	464	464
O	420	420	420	420	420	420	420	464	464	464
A	380	380	380	380	380	380	420	420	464	464
D	352	352	352	352	352	380	400	400	400	400
%	352	352	352	352	352	352	352	352	352	352
	352	352	352	352	352	352	352	352	352	352
	352	352	352	352	352	352	352	352	352	352
	352	352	352	352	352	352	352	352	352	352

Table 24: Injector Timing

The **Injector Timing Table** is expressed in crank degrees and is modified by rpm and load. The A9L values are located at 7160 hex in a 10 x 8 table, each data value being one unsigned byte. The value stored has the two LSB places shifted off so that the binary value must be multiplied by 4 to recover the data (i.e. the minimum resolution is 4 crank degrees and, I suppose, the maximum value possible is 2048).

Injector to Fire

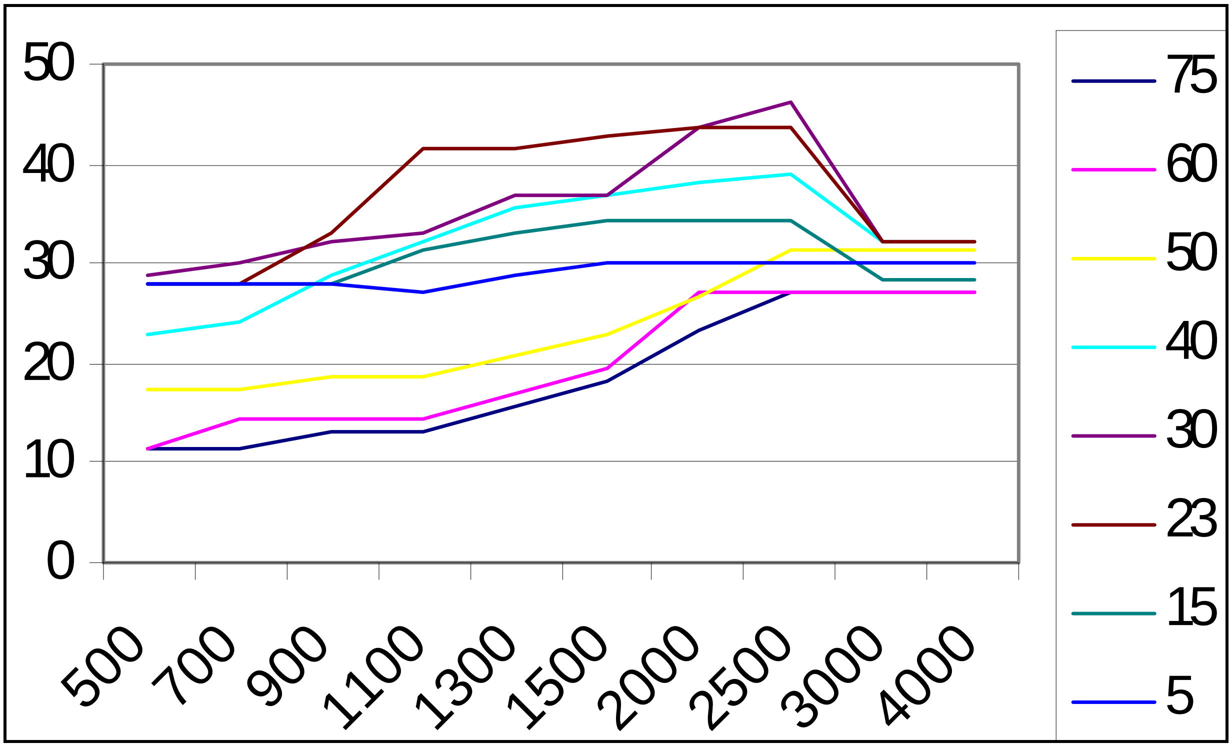
1	3	7	2	6	5	4	8

Table 25: Injector Firing Order

The **Load Injector Firing Order Table** is stored at hex 71B9 in the A9L and is the 302 firing order. It is an 8 x 1, one byte, unsigned binary value.

	RPM									
LOAD %	500	700	900	1100	1300	1500	2000	2500	3000	4000
75	11.50	11.50	13.00	13.00	15.50	18.25	23.50	27.25	27.25	27.25
60	11.50	14.25	14.25	14.25	17.00	19.50	27.25	27.25	27.25	27.25
50	17.25	17.25	18.50	18.50	20.75	23.00	26.50	31.25	31.25	31.25
40	23.00	24.25	28.75	32.25	35.75	37.00	38.00	39.00	32.25	32.25
30	28.75	30.00	32.25	33.25	37.00	37.00	43.75	46.00	32.25	32.25
23	28.00	28.00	33.00	41.50	41.50	42.75	43.75	43.75	32.25	32.25
15	28.00	28.00	28.00	31.25	33.25	34.25	34.25	34.25	28.25	28.25
05	28.00	28.00	28.00	27.25	29.00	30.00	30.00	30.00	30.00	30.00

Table 26: Base Spark [Deg BTDC]



The **Base Spark Timing Table** is stored at 7574 (30068 decimal) as unsigned byte values. Divide by 4 to convert to decimal.

	RPM									
LOAD %	500	700	900	1100	1300	1500	2000	2500	3000	4000
75	11.50	11.50	13.00	13.00	14.25	17.00	27.25	27.25	27.25	27.25
60	11.50	11.50	13.00	13.00	14.25	18.25	27.25	27.25	27.25	27.25
50	17.25	18.50	18.50	18.50	20.75	23.00	30.00	31.25	31.25	31.25
40	23.00	24.25	28.75	37.00	39.00	39.00	38.00	37.00	32.25	32.25
30	28.75	30.00	34.50	42.75	43.75	41.5	41.5	43.75	32.25	32.25
23	28.00	28.00	33.00	42.75	42.75	43.75	43.75	43.75	32.25	32.25
15	28.00	28.00	28.00	32.25	34.25	35.25	35.25	34.25	28.25	28.25
05	28.00	28.00	28.00	27.25	29.00	30.00	30.00	30.00	30.00	30.00

Table 27: Altitude Base Spark [Deg BTDC]

The **Altitude Base Spark [Deg BTDC]** table lives at 75C4 (30148 decimal) as one-byte, unsigned values. Divide them by 4 to convert to decimal.

RPM

		500	700	900	1100	1300	1500	2000	2500	3000	4000
	75	9.25	9.25	13.00	13.00	15.50	18.25	23.50	23.50	23.50	23.50
L	60	9.25	11.50	13.00	13.00	15.50	18.25	23.50	23.50	23.50	23.50
O	50	9.00	10.25	11.50	12.75	15.00	19.75	25.50	25.50	25.50	25.50
A	40	16.00	16.00	16.00	16.00	18.50	24.25	31.25	31.25	31.25	31.25
D	30	23.00	24.25	23.00	27.50	35.75	37.00	38.00	38.00	38.00	38.00
	23	25.00	26.00	30.50	38.75	40.25	42.00	43.75	43.75	43.75	43.75
%	15	28.00	28.00	28.00	31.25	33.25	34.25	34.25	34.25	34.25	34.25
	05	28.00	28.00	28.00	27.25	29.00	30.00	30.00	30.00	30.00	30.00

Table 28: Limp Mode Spark Table [Deg BTDC]

The Limp Mode Spark Table [Deg BTDC] Table lives at 7614 (30228 decimal) as unsigned byte values that must be divided by 4 to convert to their real values.

0	0	0	0	2	2	2	2

Table 29: Injector Output Port Table

The Injector Output Port Table lives at (29105 decimal) for the A9L as unsigned bytes.

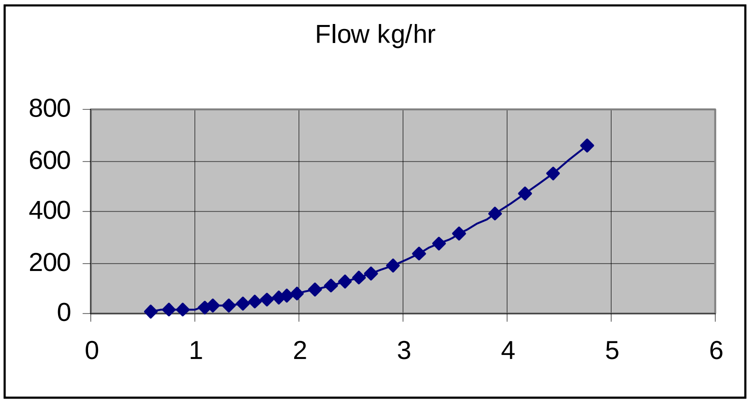
Raw Load	Table Row Pointer
0	0
0.04998779296875	0
0.100006103515625	1
0.600006103515625	6
0.75	7
1.99996948242188	7

Table 30: Load Scaling

The **Load Scaling Function** parameters for the A9L are located at 6C62. The table is 24 bytes in size, organized in a 2-byte by 2-byte unsigned, interleaved format. The “X” values (Load) must be divided by 32768 to convert them and the “Y” values (Row

Pointer) are divided by 256.

MAF Voltage	Flow kg/hr
0	10.75
0	10.75
0	10.75
0	10.75
0.571044921875	10.75
0.75	14.75
0.884033203125	18.75
1.091064453125	25.75
1.181884765625	29.00
1.321044921875	35.25
1.468017578125	42.75
1.573974609375	49.25
1.69091796875	57.50
1.802978515625	65.00
1.889892578125	71.25
1.989013671875	79.00
2.154052734375	94.75
2.31591796875	112.25
2.449951171875	127.25
2.58203125	144.00
2.694091796875	159.25
2.909912109375	192.00
3.14892578125	233.25
3.35498046875	274.25
3.5439453125	315.00
3.885986328125	394.25
4.177001953125	470.25
4.443115234375	548.75
4.76806640625	659.25
15.999755859375	659.25



MAF Transfer Function

Table 31: MAF Transfer Function

The **MAF Transfer Function** is located at 6DE2 for the A9L, is 120 bytes in length, unsigned, and is 2-byte/4096 “X” (MAF voltage) by 2-byte/4 (mass air flow in kg/hr).

RPM	Advance
0.00	10.25
1000.00	10.25
1800.00	22.25
2150.00	25.75
2600.00	26.75
3500.00	26.75
5000.00	30.25
16383.75	30.25

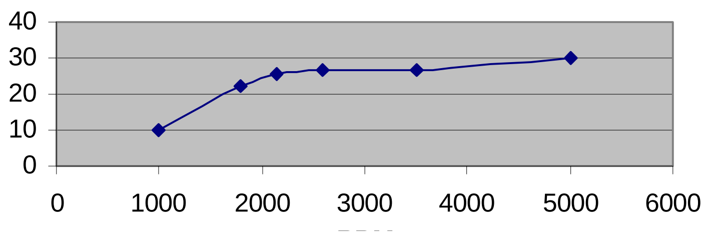


Table 32: WOT Spark Advance vs RPM

The **WOT Spark Advance vs. RPM function** is located at 7504 hex for the A9L and is composed of 8 interleaved X-Y values, each 2 bytes in length, each unsigned and each in two binary point format (divide the value by 4 to get a resolution of .25). The advance is in degrees BTDC.

ECT	Advance
-256	2
86	3
120	0
200	0
236	-2
246	-4
254	-4

Table 33: WOT Spark Advance vs. ECT

The **WOT Spark Advance vs. ECT Function** is located at 7532 hex in the A9L EEC. An interleaved table of 7 one-byte ***signed*** X-Y values, the “X” values (ECT) are multiplied by 2 and the “Y” values (advance inn degrees BTDC) are divided by 4 to convert them.

ACT	Advance
-256	0
-256	0
-256	0
150	0
240	-6
254	-6

Table 34: WOT Spark Advance vs. ACT

The **WOT Spark Advance vs. ACT Function** is located at 7540 hex in the A9L in a signed, one-byte format for both the X and the Y interleaved values. The “X” value (ACT) is multiplied by 2 and the “Y” value (Advance in degrees BTDC) is divided by 4.

TP (A/D counts)	Enrichment Multiplier
0	1
200	0.953125
452	0.296875
852	0
1020	0

Table 35: Accelerator Enrichment Multiplier vs TP

The **Accelerator Enrichment Multiplier vs. TP Function**, located at 71C1 in the A9L, is a table of interleaved X-Y values each unsigned and one byte in length. To convert them to decimal, the “X” values (TP sensor in A-D counts) must be multiplied by 4 and the “Y” values (the fuel enrichment multiplier) must be divided by 128.

RPM	Multiplier
0	1
1600	0.90625
2400	0.96875
3200	0.9453125
3800	0.9765625
4400	0.953125
6000	0.9453125

RPM	Multiplier
16383.75	0.9453125

Table 36: WOT Fuel Multiplier vs RPM

The **WOT Fuel Multiplier vs. RPM Function** parameters are stored at hex location 7680 in the A9L. They are unsigned, 2 bytes in length and are arranged in an interleaved format, with the “X” values (RPM) needing to be divided by 4 to convert them to decimal and the “Y” values (fuel multiplier) needing to be divided by 128.

Intake Air Charge Temp	Fuel Multiplier
-256	0.640625
-256	0.640625
-40	0.640625
0	0.812500
76	1.000000
254	1.000000

Table 37: Open Loop Fuel Multiplier vs. ACT

The **Open Loop Fuel Multiplier vs. ACT Function** parameters are located at 7B48. They are single byte values stored as alternating X and Y values. The “X” values (intake air charge temperature) are signed and have to be multiplied by two to convert them to decimal. The “Y” values (open loop fuel multiplier) are unsigned and have to be divided by 64.

Intake Air Charge Temp	Spark Advance
-256	0
-256	0
160	0
190	-2
254	-2

Table 38: Spark Advance vs. ACT

The **Spark Advance vs. ACT Function** parameters, for the A9L, are located at 74EE. It consists of single-byte signed values. The “X” values (intake air charge temperature) must be multiplied by 2 to convert them to decimal and the “Y” values must be divided by 4.

Barometric Pressure	Spark Adder Deg
0	12
23.500	4
26.500	4
27.750	0
31.875	0

Table 39: Spark Advance vs. BP

The Spark Advance vs. BP Function parameters are located at 7528 and consist of signed, single-byte values. Divide the “X” values (barometric pressure) by 8 to convert them to decimal and the “Y” values (spark degrees to add to the timing) nneed to be divided by 4.

RPM	Seconds/ Degree
0	2.0000
0	2.0000
992	2.0000
3008	1.5000
4000	1.0000
8160	1.0000

Table 40: Spark Advance Rate vs. RPM

The Spark Advance Rate vs. RPM Function parameters for the A9L are located at 779C, are one byte values, with the “X” values (RPM) being signed and needing to be multiplied by 32 to convert them to decimal. The “Y” values (seconds/degree) are unsigned and must be divided by 16.

Battery Voltage	Dwell
0.0000	589.75
10.0000	589.75
12.0000	524.25
14.0000	393.25
15.0000	367.00
4095.9375	367.00

Table 41: Minimum Low Speed Dwell

Minimum Low Speed Dwell is a function whose parameters are stored at 7748 as unsigned 16-bit values. To convert them to decimal the “X” values (battery voltage) must be divided by 16 and the “Y” values (dwell / coil OFF time in milliseconds) must be divided by 4.

Battery Voltage	Dwell
0.0000	655.25
10.0000	655.25
12.0000	485.00
14.0000	439.00
15.0000	393.25
4095.9375	393.25

Table 42: Minimum High Speed Dwell

The **Minimum High Speed Dwell Function** parameters are located at 7760. They are stored as alternating X and Y values in an unsigned, 16-bit format. To convert them to decimal, divide the “X” values (battery voltage) by 16 and the “Y” values (dwell / coil off time in mS) by 4.

RPM	Dashpot Clip (lb/min)
0.00	0.12011718750
500.00	0.12011718750
1500.00	1.00000000000
2000.00	1.60009765625
16383.75	1.60009765625

Table 43: Maximum Dashpot Clip

The **Maximum Dashpot Clip Function** parameters are stored at 7346 as unsigned, 16-bit, intertwined X and Y values. To convert them to decimal, divide the the RPM values by 4 and the dashpot clip values by 4096.

RPM	Fuel Multiplier
0.00	1.000000
0.00	1.000000
650.00	1.000000
750.00	1.203125
1200.00	1.203125
1500.00	1.000000
16383.75	1.000000

Table 44: Sea Level Lugging O/L Fuel Multiplier

The **Sea Level Lugging O/L Fuel Multiplier Function** parameters are stored at 782A as unsigned, 16-bit numbers. To convert them to decimal, divide the “X” values (RPM) by 4 and the “Y” values (Lugging O/L Fuel Multiplier) by 128.

RPM	Fuel Multiplier
0.00	1
0.00	1
0.00	1
0.00	1
0.00	1
0.00	1
16383.75	1

Table 45: Altitude Lugging O/L Fuel Multiplier

The Altitude Lugging O/L Fuel Multiplier Function parameters are stored at 7846 as 16-bit, unsigned words. To convert them to decimal, divide the “X” values (RPM) by 4 and the “Y” values (Lugging O/L Fuel Multiplier) by 128.

Time in Crank Mode	Crank Fuel Multiplier
0.000	1
0.000	1
0.000	1
0.000	1
0.000	1
8191.875	1

Table 46: Crank Fuel Pulse Width Multiplier

The **Crank Fuel Pulse Width Multiplier Function** parameters live at 729A as unsigned 16-bit words. Divide the “X” values by 8 and the “Y” values by 128.

ECT	Pulse Width (mS)
-65536	39.69140625
-20	39.69140625
0	22.41406250
20	14.48828125
40	10.75390625
70	5.83593750

Table 47: Cranking Fuel Pulse width vs. ECT

ECT	Pulse Width (mS)
150	2.35546875
180	2.04687500
65534	2.04687500

Table 47: Cranking Fuel Pulse width vs. ECT

The Cranking Fuel Pulse width vs. ECT Function parameters are stored beginning at 787A as 16-bit words. The “X” values (ECT) are signed and must be multiplied by 2 to convert them to decimal. The “Y” values (pulse width in milliseconds) are unsigned and must be divided by 256.

Battery Voltage	Injector Offset
0	7.50000
5.5000	7.50000
6.0000	5.53125
7.0000	5.53125
8.0000	2.59375
9.0000	2.03125
10.0000	1.68750
11.0000	1.40625
12.0000	1.15625
13.0000	1.00000
14.0000	0.90625
15.9375	0.90625

Table 48: Injector Offset vs. Battery Voltage

The **Injector Offset vs. Battery Voltage Function** parameters reside at 7B30 as unsigned, single byte values. The “X” values (Battery Voltage) have to be divided by 16 to convert them to decimal and the “Y” values (Injector Offset) by 32.

RPM	Open Loop Fuel Multiplier
0	1
0	1
0	1
1075.00	1
1100.00	0
16383.75	0

Table 49: Open Loop Fuel Multiplier vs. RPM

The **Open Loop Fuel Multiplier vs. RPM Function** parameters live at 7DE8 as unsigned, 16-bit words. Divide the “X” values (RPM) by 4 to convert them to decimal and the “Y” values (Open Loop Fuel Multiplier) by 128.

Dashpot[lb/min]	Decrement Rate [lb/min]
0	0.003417968750
0.0620117187500	0.002929687500
0.3500976562500	0.001953125000
0.4499511718750	0.007080078125
0.6499023437500	0.044921875000
0.7500000000000	0.062500000000

Dashpot[lb/min]	Decrement Rate [lb/min]
15.999755859375	0.062500000000

Table 50: Dashpot Decrement Rate Function

The Dashpot Decrement Rate Function parameters live at 732A as unsigned 16-bit words. To convert them to decimal, divide both the “X” values (Dashpot[lb/min]) and the “Y” values (Decrement Rate [lb/min]) by 4096.

Data Pointers

There are 8 pointers at 2022h:

```
V2020:  DW    L0108
        DW    L8C00     ;2022
        DW    L8E5A     ;2024
        DW    L8FD6     ;2026....
        DW    L93A8
        DW    L9430
        DW    L97A8
        DW    L9A7E
        DW    L9E5A
```

Notice the byte at 2020h is 8 and there are 8 pointers.

At each address there is a pointer to the next - a linked list...

```
L8C00:  DW  08E5AH

        functions/maps/scalars
L8E5A:  DW  08FD6H

        functions/maps/scalars
L8FD6:  DW  093A8H

        functions/maps/scalars
L93A8:  DW  09430H

        functions/maps/scalars
L9430:  DW  097A8H

        functions/maps/scalars
L97A8:  DW  09A7EH

        functions/maps/scalars
L9A7E:  DW  09E5AH

        functions/maps/scalars
L9E5A:  DW  09EA2H

        ;ORG09EA2H
        FILL0FFH,0015EH
        ;end
```

Idle Speed Constants

These are the idle speed locations for the A9L 5.0 stang MAF.

Parameter	Value	Address	Notes
idle in neutral	672 RPM	704A	0054H = 84 and 84 * 8 = 672
idle in gear	672	704C	

Table 51: A9L Idle Speed Addresses

Conversion:

(Note that, in Intel format, the LSB is first, so it’s actually as "0054", and $54_{16} = 84_{10}$. Multiply by 8 to get 672_{10}.

Speed Limiters

```
ROM addr : Data
0x7A86   : FF
0x7A87   : FF
0x7A88   : FF
0x7A89   : FF
```

There are two speed limiters here, and an ON and OFF value (With hysteresis, I think. Which means once you hit the speed limiter you need to slow down to the off value before you can accelerate again)

When data = FF the speed limiter is off, otherwise the conversion is as follows:

0FFH --> 255, then divide by 2 and you get 127.5

Not sure which is on and which is off. Maybe someone else can help.

I think it’s

```
Speed Limiter A: 0x7A86 ON; 0x7A87 OFF
Speed Limiter B: 0x7A88 ON; 0x7A89 OFF
```

The ONs and OFFs may be swapped!

Rev Limiter

Max RPMs seems to be PIP filter related! Any thoughts here? Please post your ideas to the EEC group.

At address 0x7C16 I have for data C103 (reverse bytes to get hi-byte first, 03C1). H03C1 --> D961 which I think is the PIP filter. When I cut that number in half, my rev limit is half, so it definitely affects the rev limit.

Half Fuel Limits

They aren’t used in the stang, but if you wanted a half fuel limit:

ROM addr : Data

```
0x7A82   : 007D   half fuel ON
0x7A84   : 307F   half fuel OFF

007D --> 7D00 (little endian, remember)
H7D00 = D32000 divide by 4 to get 8,000 RPMS for ON

307F --> 7F30 (little endian, remember)
H7F30 = D32560 divide by 4 to get 8,140 RPMS for OFF
```

Not only is this programmed above the "MAX" limit, OFF is higher than ON.

A9L Tables

(Gary Anderson wrote) "Just so you guys remember where Tom got the data:"

```
a9l.v5.out
"Accel Enrichment Fuel [lb/min]
```

```
"Throttle Rate [Deg/sec] vs ECT
"table:29131,8,7,1,d4,0
:1.5,1.5,2,3,4.5,8,12,18
:1.5,1.5,2,3,4.5,8,12,20
:2,2,3,4.5,8,12.5,18,29
:4,5,6,8,12.5,17,25.5,34
:5,8,10,12,16.5,21,27,45
:12,14,16,18,26,30,38,55
:21,23,26,28,31,34,39,56
"end
"Startup Fuel [A/F Ratio]
"ECT vs. Time
"table:31808,10,8,1,d8,0
:0,0,0,0,0,0,0,0,0,0
:1,1.25,0.875,0.5,0.375,0.875,0.875,0.25,0.5,1
:1.25,1.5,1.125,1,1,1.25,1.25,0.875,1,1.25
:1.625,1.625,1.5,1.375,1.25,1.375,1.375,1.125,1.375,1.625
:2.25,2.125,2.125,1.875,1.875,1.875,1.875,1.25,1.875,2.25
:2.75,2.75,2.375,1.875,1.875,1.875,1.875,1.5,1.875,2.75
:2.875,3.625,2.625,1.875,1.875,1.875,1.875,1.625,1.875,2.875
:3.375,4.125,3,1.875,1.875,1.875,1.875,1.375,1.875,3.375
"end
"Base Fuel [A/F Ratio]
"ECT vs. Load
"table:31728,10,8,1,d8,0
:10.5,10.5,12.5,12.5,12.75,13.125,13.5,13.5,13.5,13.5
:11.25,11.25,12.75,12.75,13.125,13.5,13.75,13.75,13.625,13.5
:12.125,12.125,13.5,13.5,13.875,14.375,14.125,14.25,13.625,13.625
:12.75,13.875,14.25,14.375,14.875,14.875,15.25,16,16,15
:13.625,14.375,14.375,14.375,14.875,15.25,15.25,16,16,14.75
:14.125,14.25,14.25,14.375,14.5,14.75,14.75,15.25,15.25,14.375
:17.625,17.625,17,16.625,16.5,16.375,16,15.25,15.25,15.25
:17.875,17.875,17.625,17.25,16.625,16.375,16,15.25,15.25,15.25
"end
"Injector Timing [Crank Degrees]
"RPM vs. Load
"table:29024,10,8,1,m4,0
:420,420,420,420,420,420,420,464,464,464
:420,420,420,420,420,420,420,464,464,464
:380,380,380,380,380,380,420,420,464,464
:352,352,352,352,352,380,400,400,400,400
:352,352,352,352,352,352,352,352,352,352
:352,352,352,352,352,352,352,352,352,352
:352,352,352,352,352,352,352,352,352,352
:352,352,352,352,352,352,352,352,352,352
"end
"Load Injector Firing Order
"Injector to Fire vs.
"table:29113,8,1,1,1,0
:1,3,7,2,6,5,4,8
"end
"Base Spark [Deg BTDC]
"RPM vs. Load
"table:30068,10,8,1,d4,0
:11.5,11.5,13,13,15.5,18.25,23.5,27.25,27.25,27.25
:11.5,14.25,14.25,14.25,17,19.5,27.25,27.25,27.25,27.25
:17.25,17.25,18.5,18.5,20.75,23,26.5,31.25,31.25,31.25
:23,24.25,28.75,32.25,35.75,37,38,39,32.25,32.25
:28.75,30,32.25,33.25,37,37,43.75,46,32.25,32.25
:28,28,33,41.5,41.5,42.75,43.75,43.75,32.25,32.25
:28,28,28,31.25,33.25,34.25,34.25,34.25,28.25,28.25
:28,28,28,27.25,29,30,30,30,30,30
"end
"Altitude Base Spark [Deg BTDC]
"RPM vs. Load
"table:30148,10,8,1,d4,0
:11.5,11.5,13,13,14.25,17,27.25,27.25,27.25,27.25
:11.5,11.5,13,13,14.25,18.25,27.25,27.25,27.25,27.25
```

```
:17.25,18.5,18.5,18.5,20.75,23,30,31.25,31.25,31.25
:23,24.25,28.75,37,39,39,38,37,32.25,32.25
:28.75,30,34.5,42.75,43.75,41.5,41.5,43.75,32.25,32.25
:28,28,33,42.75,42.75,43.75,43.75,43.75,32.25,32.25
:28,28,28,32.25,34.25,35.25,35.25,34.25,28.25,28.25
:28,28,28,27.25,29,30,30,30,30,30
"end
"Limp Mode Spark Table [Deg BTDC]
"RPM vs. Load
"table:30228,10,8,1,d4,0
:9.25,9.25,13,13,15.5,18.25,23.5,23.5,23.5,23.5
:9.25,11.5,13,13,15.5,18.25,23.5,23.5,23.5,23.5
:9,10.25,11.5,12.75,15,19.75,25.5,25.5,25.5,25.5
:16,16,16,16,18.5,24.25,31.25,31.25,31.25,31.25
:23,24.25,23,27.5,35.75,37,38,38,38,38
:25,26,30.5,38.75,40.25,42,43.75,43.75,43.75,43.75
:28,28,28,31.25,33.25,34.25,34.25,34.25,34.25,34.25
:28,28,28,27.25,29,30,30,30,30,30
"end
"Injector Output Port
"Injector Port vs.
"table:29105,8,1,1,1,0
:0,0,0,0,2,2,2,2
"end
"EEC Load Scaling
"func:27826,24,2,d32768,0,2,d256,0
Raw Load:0,0.04998779296875,0.100006103515625,0.600006103515625,0.75,1.99996948242188
Table Row Pointer:0,0,1,6,7,7
"end
"MAF Transfer
"func:28130,120,2,d4096,0,2,d4,0
MAF
Voltage:0,0,0,0,0.571044921875,0.75,0.884033203125,1.091064453125,1.181884765625,1.321044921875,1.468
017578125,1.573974609375,1.69091796875,1.802978515625,1.889892578125,1.989013671875,2.154052734375,2.
31591796875,2.449951171875,2.58203125,2.694091796875,2.909912109375,3.14892578125,3.35498046875,3.543
9453125,3.885986328125,4.177001953125,4.443115234375,4.76806640625,15.999755859375
Mass Flow Kg/
hr:10.75,10.75,10.75,10.75,10.75,14.75,18.75,25.75,29,35.25,42.75,49.25,57.5,65,71.25,79,94.75,112.25
,127.25,144,159.25,192,233.25,274.25,315,394.25,470.25,548.75,659.25,659.25
"end
"WOT Spark Advance vs. RPM
"func:29956,32,2,d4,0,2,d4,1
RPM:0,1000,1800,2150,2600,3500,5000,16383.75
WOT Spark Advance Deg BTDC:10.25,10.25,22.25,25.75,26.75,26.75,30.25,30.25
"end
"WOT Spark Advance vs. ECT
"func:30002,14,1,m2,1,1,d4,1
Engine Coolant Temp:-256,86,120,200,236,246,254
WOT Spark Advance Deg BTDC:3,3,0,0,-2,-4,-4
"end
"WOT Spark Advance vs. ACT
"func:30016,12,1,m2,1,1,d4,1
Intake Air Charge Temp:-256,-256,-256,150,240,254
WOT Spark Advance Deg BTDC:0,0,0,0,-6,-6
"end
"Accelerator Enrichment Multiplier vs. TP
"func:29121,10,1,m4,0,1,d128,0
Throttle Position-A/D Counts:0,200,452,852,1020
Accelerator Enrichment Multiplier:1,0.953125,0.296875,0,0
"end
"WOT Fuel Multiplier vs. RPM
"func:31616,32,2,d4,0,2,d128,0
RPM:0,1600,2400,3200,3800,4400,6000,16383.75
WOT Fuel Multiplier:1,0.90625,0.96875,0.9453125,0.9765625,0.953125,0.9453125,0.9453125
"end
"Open Loop Fuel Multiplier vs. ACT
"func:31560,12,1,m2,1,1,d64,0
Intake Air ChargeTemp:-256,-256,-40,0,76,254
```

```
Open Loop Fuel Multiplier:0.640625,0.640625,0.640625,0.8125,1,1
"end
"Spark Advance vs. ACT
"func:29934,10,1,m2,1,1,d4,1
Air Charge Temperature:-256,-256,160,190,254
Spark Advance [Deg BTDC]:0,0,0,-2,-2
"end
"Spark Advance vs. BP
"func:29992,10,1,d8,0,1,d4,1
Barometric Pressure:0,23.5,26.5,27.75,31.875
Spark Adder Deg:12,4,4,0,0
"end
"Spark Advance Rate vs. RPM
"func:30620,12,1,m32,0,1,d16,0
RPM:0,0,992,3008,4000,8160
Seconds/Degree:2,2,2,1.5,1,1
"end
"Min Low Speed Dwell
"func:30536,24,2,d16,0,2,d4,0
Battery Voltage:0,10,12,14,15,4095.9375
Dwell / Coil Off Time [ms]:589.75,589.75,524.25,393.25,367,367
"end
"Min High Speed Dwell
"func:30560,24,2,d16,0,2,d4,0
Battery Voltage:0,10,12,13,15,4095.9375
Dwell / Coil Off Time [ms]:655.25,655.25,485,439,393.25,393.25
"end
"Maximum Dashpot Clip
"func:29510,20,2,d4,0,2,d4096,0
RPM:0,500,1500,2000,16383.75
Dashpot Clip [lb/min]:0.1201171875,0.1201171875,1,1.60009765625,1.60009765625
"end
"Sea Level Lugging O/L Fuel Multiplier
"func:30762,28,2,d4,0,2,d128,0
RPM:0,0,650,750,1200,1500,16383.75
Lugging O/L Fuel Multiplier:1,1,1,1.203125,1.203125,1,1
"end
"Altitude Lugging O/L Fuel Multiplier
"func:30790,28,2,d4,0,2,d128,0
RPM:0,0,0,0,0,0,16383.75
Lugging O/L Fuel Multiplier:1,1,1,1,1,1,1
"end
"Crank Fuel Pulse Width Multiplier
"func:29338,24,2,d8,0,2,d128,0
Time in Crank Mode:0,0,0,0,0,8191.875
Crank Fuel Multiplier:1,1,1,1,1,1
"end
"Cranking Fuel Pulse width vs. ECT
"func:30842,36,2,m2,1,2,d256,0
Engine Coolant Temperature:-65536,-20,0,20,40,70,150,180,65534
Pulse width
[ms]:39.69140625,39.69140625,22.4140625,14.48828125,10.75390625,5.8359375,2.35546875,2.046875,2.04687
5
"end
"Injector Offset vs. Battery Voltage
"func:31536,24,1,d16,0,1,d32,0
Battery Voltage:0,5.5,6,7,8,9,10,11,12,13,14,15.9375
InjectorOffset:7.5,7.5,5.53125,3.53125,2.59375,2.03125,1.6875,1.40625,1.15625,1,0.90625,0.90625
"end
"Open Loop Fuel Multiplier vs. RPM
"func:32232,24,2,d4,0,2,d128,0
RPM:0,0,0,1075,1100,16383.75
Open Loop Fuel Multiplier:1,1,1,1,0,0
"end
"Dashpot Decrement Rate
"func:29482,28,2,d4096,0,2,d4096,0
Dashpot[lb/min]:0,0.06201171875,0.35009765625,0.449951171875,0.64990234375,0.75,15.999755859375
Decrement Rate [lb/
```

```
    min]:0.00341796875,0.0029296875,0.001953125,0.007080078125,0.044921875,0.0625,0.0625
    "end
```

A9L Table Format Explanation

(Bill Lawrance wrote)

I’m putting this note in front of Gary Anderson’s recent post on tables for the A9L. I hope it helps in understanding them. Gary has done a great piece of work here. He deserves real credit for digging this out. Everything I’ve added here I got from studying his post.

This note describes how the tables displayed in the post are represented in the EPROM. I got this from reading the program. This is only useful or interesting if you want to see how this is done. Nothing in the original post or in this note tells how these tables are actually used, although their titles are descriptive. Neither are there any units on the axes of the tables, although their titles are also descriptive.

There are 2 types of tables displayed.

----

1) Two dimension Tables

The first of the two kinds of tables is identified in the output with ’table’ in the third line of output for a table. For example, the first table listed is:

```
"Accel Enrichment Fuel [lb/min]
"Throttle Rate [Deg/sec] vs. ECT
"table:29131,8,7,1,d4,0
:1.5,1.5,2,3,4.5,8,12,18
:1.5,1.5,2,3,4.5,8,12,20
:2,2,3,4.5,8,12.5,18,29
:4,5,6,8,12.5,17,25.5,34
:5,8,10,12,16.5,21,27,45
:12,14,16,18,26,30,38,55
:21,23,26,28,31,34,39,56
"end
```

--The first line is the title of the table

--The second line is "X-axis label vs. Y-axis label"

--The third line is decoded as follows:

```
table   -- identifies this as a 2 dimension table
29131   -- the decimal location of the table in the PROM (or 71CB hex )
    8   -- the number of columns (X-axis) in the table
    7   -- the number of rows    (Y-axis) in the table
    1   -- number of bytes for each value in the table
```

d4 -- the scale of the value, in this case divide by 4 to get lb/min of fuel that each entry represents. ’m’ is used in other tables to specify ’multiply each entry’.

```
    0   -- each entry is unsigned
```

From the binary:

```
0071C0                         -         54 5C 68 70 7C
0071D0  88 9C E0 30 38 40 48 68-78 98 DC 14 20 28 30 42
0071E0  54 6C B4 10 14 18 20 32-44 66 88 08 08 0C 12 20
0071F0  32 48 74 06 06 08 0C 12-20 30 50 06 06 08 0C 12
007200  20 30 48
```

Grouping into rows of 8 columns:

```
54 5C 68 70 7C 88 9C E0
30 38 40 48 68 78 98 DC
14 20 28 30 42 54 6C B4
10 14 18 20 32 44 66 88
08 08 0C 12 20 32 48 74
06 06 08 0C 12 20 30 50
06 06 08 0C 12 20 30 48
```

Converting to decimal and dividing by 4:

```
                    21  23  26  28  31  34  39  56
                    12   ...
                     5   ...
                     4   ...
                     2   ...
                   1.5 1.5   2   3 4.5   8  12  20
                   1.5 1.5   2   3 4.5   8  12  18
```

The rows are stored last first, so reversing rows gives the table as displayed:

```
                   1.5 1.5   2   3 4.5   8  12  18
                   1.5 1.5   2   3 4.5   8  12  20
                     2   ...
                     4   ...
                     5   ...
                    12   ...
                    21  23  26  28  31  34  39  56
```

----

2) Function Tables

The second of the two kinds of tables is identified in the output with 'func' in the second line of output for it. For example, the 'WOT Spark Advance vs. RPM' table is:

```
"WOT Spark Advance vs. RPM
"func:29956,32,2,d4,0,2,d4,1
RPM:0,1000,1800,2150,2600,3500,5000,16383.75
WOT Spark Advance Deg TDC:10.25,10.25,22.25,25.75,26.75,26.75,30.25,30.25
"end
```

--The first line is the title of the table

--The second line is decoded as follows:

```
 func   -- identifies this as a function type table
29956   -- the decimal location of the table in the PROM
              (or  7504 hex)
   32   -- the total number of bytes in the table
    2   -- the number of bytes in each X entry
   d4   -- the scale of each X entry. In this case, divide
               each X entry by 4 to get RPM
               'm' can be used to specify 'multiply each X entry'.
    0   -- each X entry is unsigned
    2   -- the number of bytes in each Y entry
   d4   -- the scale of each Y entry. In this case, divide
               each Y entry by 4 to get degrees of Spark Advance
               'm' can be used to specify 'multiply each Y entry'.
    1   -- each Y entry is signed
```

From the binary:

```
007500              FF FF 79 00-20 4E 79 00 B0 36 6B 00
007510  A0 28 6B 00 98 21 67 00-20 1C 59 00 A0 0F 29 00
007520  00 00 29 00
```

Since each entry is 2 bytes long, group into pairs:

```
FF FF                                  FF FF
79 00                                  00 79
20 4E                                  4E 20
79 00                                  00 79
B0 36                                  36 B0
6B 00   Since this is an Intel --->    00 6B
A0 28   CPU, reverse bytes of  --->    28 A0
6B 00   each integer.          --->    00 6B
98 21                                  21 98
67 00                                  00 67
20 1C                                  1C 20
59 00                                  00 59
```

```
                  A0 0F                                  0F A0
                  29 00                                  00 29
                  00 00                                  00 00
                  29 00                                  00 29
```

Converting to decimal and dividing by 4 gives:

```
            16383.75
               30.25
             5000.
               30.25
             3500.
               26.75
             2600.
               26.75
             2150.
               25.75
             1800.
               22.25
             1000.
               10.25
                0.
               10.25
```

In this type of table, the X and Y values are interleaved and stored in reverse. So, rearranging gives the table as displayed:

```
           X-Values   0     1000   1800   2150   2600   3500   5000  16383.75
           Y-Values  10.25  10.25  22.25  25.75  26.75  26.75  30.25  30.25
```

DA1 Constants & Locations

The DA-1 EEC fuel table is at 7ABE, 10,8 size. WOT ignition 7CC0, 4,8 size. Part throttle tables 7CE0,7D30,7F08, 10,8 in size.

SENSORS

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

EGO

Can the HEGO respond quickly enough to monitor each cylinder during engine operation? Cylinder-by-cylinder HEGO function can sometimes be seen during testing on engine dynamometers, but only at certain points. HEGO response time is 300 to 1500 milliseconds for a delta Lambda step of 0.1, either rich-to-lean, or lean-to-rich. Driven into rich or lean saturation, the sensor can require several seconds to recover. Assuming best case 300 milliseconds response time, in a V8 firing 2 cylinders per crankshaft revolution with stereo EGO’s, the highest RPM one could hope to discern individual exhaust pulses would be about 300. Getting this information from the HEGO into EEC is also made difficult by the low pass filtering on the HEGO inputs, which are typically -3dB at 1 to 20 Hz, application dependent. This is to filter out HEGO buzz, which is the rapid output oscillation that HEGO’s produce when at stoichiometry.

Another problem that cylinder-by-cylinder mixture correction would create is with closed loop control bandwidth: An engine is basically an air pump with a finite response time to external stimulus, such as moving the throttle plate. Let’s say it’s about 1 second. Basic control systems theory says that if you put said air pump into a closed loop control system, the control system bandwidth must be slower than the air pump response time, else the system will hunt, or be unstable.

With regards to lean operation with an O2 sensor, the O2 sensor’s output is not particularly useful far from stoich. The time-honored way of biasing the mixture is to vary the switch voltage threshold (in software), and to low-pass filter the O2 sensor signal (again in software) and try to keep its average to a calculated value. This can vary the mixture sufficiently to optimize the operation of a catalyst. A normal (heated or unheated) O2 sensor cannot tell you if the mixture is rich enough for best power or not. The O2 sensor just goes to RICH, and you can’t tell whether you are at 9:1 or 14:1 from the sensor though in the 14 to 15:1 region you may get some indication. Typically, the system extrapolates from the highest load at closed loop fuel trim value and applies this same trim value when power enrichment is on. This simple method works because injectors tend to be linear. If, at some operating point, it takes an 8 ms ’valid’ injection time (take the offset of the injector away) for a 16:1 mixture, then a 16 ms injection time would, ideally, give you an 8:1 mixture.

The L03 1989 Caprice (GM) uses a strategy called lean on cruise. This alternates five minute periods of operation at about 16.5:1 with one minute intervals of stoich operation. This allows the ECM to run lean most of the time, but correct for changes in conditions. Interestingly enough, the O2 voltage does not ’spike’ much when going from open to closed loop: the engine’s operating conditions don’t change that much.

EGR

The EGR reduces pumping loss by reducing manifold vacuum. It does, however, lower NOx by reducing the concentration of O2 in the cylinder. In essence, it lowers the oxygen content of the air by some 10 to 15%. This added ’inert’ gas absorbs the energy from burning the fuel and oxygen, releasing it on the expansion stroke. It would be equivalent to raising the nitrogen content of air from 78 or 79% up to around 82 or 85%. The more dilute oxygen tends to lower the combustion chamber temperature. The reverse is true when you shoot nitrous oxide into an engine. In that case, you raise the oxygen content of the air and, consequently, raise the combustion pressures and temperatures (and not incidentally, power).

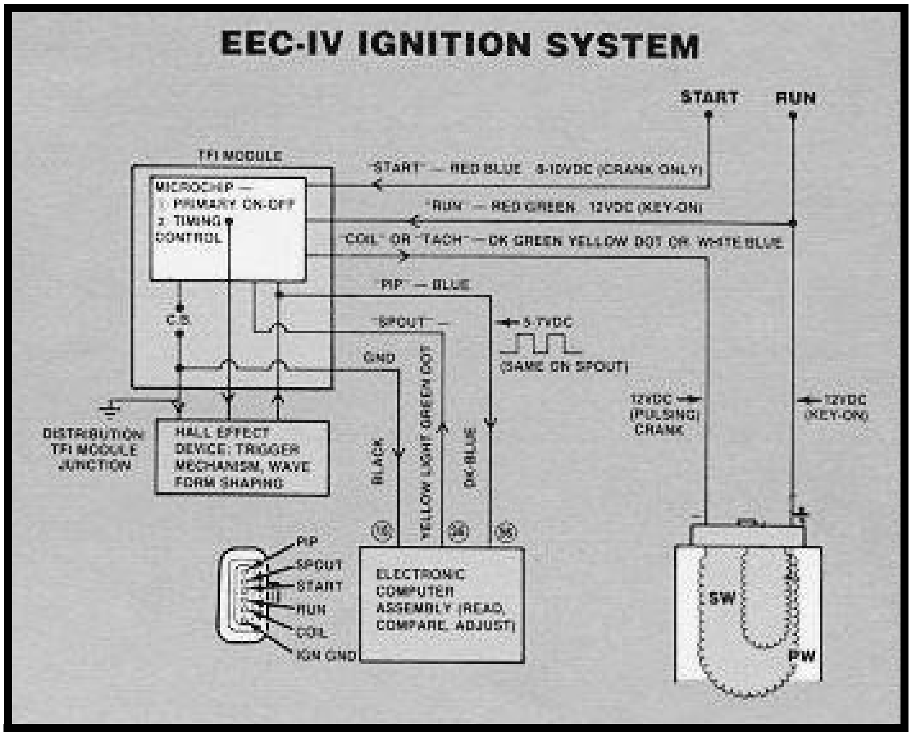
ACT

This is snagged from the graph in the Calibrator demo program. To determine air temp you can measure voltage at the ACT (air charge temp sensor). Reference is 5.0 volts.

Air Temp^{o}F	50	68	86	104	122	140	158	176	194	212	230	248
Voltage	3.52	3.06	2.62	2.16	1.72	1.35	1.04	.80	.61	.47	.36	.28

Table 52: ACT Transfer Function

TFI / PIP



Normally, the distributor contains the TFI (Thick Film Ignition) module and a hall effect pickup with a vane stator. The PIP (Profile Ignition Pickup) Signal begins when the hall-effect timing pickup switch sends a signal to the TFI module. Modification by a schmitt trigger converts this signal to a logic compatible output that is then sent to the EEC, which is responsible for

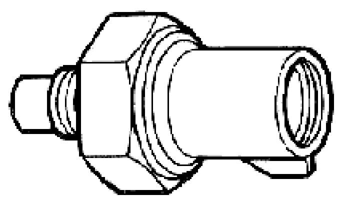
Early EEC’s used a sectored wheel (vanes) in the distributor with the number of vanes being equal to half the number of cylinders. Later EECs utilize a 36 tooth wheel, with a missing tooth for synchronization, whose output is pre-processed by a unit known as the EDIS (Electronic DIStributor). The EEC processes the PIP signal and subsequently feeds a properly advanced or retarded signal back to the TFI ignition module to fire the plug.

KNOCK SENSOR

(basic part number: 12A699)

part #	Res. f	color	usage	thread
E3AF-AA	5.4K	BLK	5.8L Ford/Mercury	1/2-13 UNC
E3ZSF-AA	5.7K	GRY	2.3L (Turbo) T’bird	M12 x 1.5mm-6g
E6TF-AA	6.45K	BLK	2.9L Ranger/Bronco II	M10 x 1.5mm-6g
E5TF-AA	6.0K	BLK	3.0L Taurus/Sable, Aerostar	M10 x 1.5mm-6g
37TF-AA	9.5K	WHT	4.9L Econoline/Bronco	M10 x 1.5mm-6g

Table 53: Knock Sensor Data



Knock Sensor

VSS

VSS gets used for many things, even in the manual gearbag calibrations.

1. Used to drive coast-down strategies. EEC will use longer throttle dashpots if it sees a non-zero signal from the VSS at closing throttle. This prevents lift-throttle snatch, especially in lower gears and at lower road speeds.

2. Drives radiator fan strategies. Fan is usually turned off above 45 MPH unless coolant temp is above fan-on setpoint.

3. Cruise control on vehicles with the cruise control integrated into the EEC, instead of a separate electronic module..

The engine should run fine without the VSS. I have disconnected the VSS in my ’89 SHO and ’98 Contour V6 to get rid of the long dashpot. If the strategy doesn’t see a VSS signal, it drops the throttle immediately, which puts ME in control of the shift rev-matching, not the bloody computer.

The cruise control is separate from the EEC, but they share the same sensor. For a manual transmission MAF Mustang, the VSS is used to maintain an 1100 idle when coasting to a stop until you drop below ~8 mph. I’ve known a few people who converted their speed density cars to MAF that had stalling problems when pushing in and holding the clutch at speed until the VSS sensor was hooked up. The speed density cars did not use the sensor. I’m not aware of the sensor being used for anything else except on the ’93 Cobra, where it was used to change the timing curve when above 100 something miles per hour.

AFM

Modern bypass MAFS have the best resolution in the first 40% of their total flow, with some resolution losses of very low flows at idle and just off-idle.

Raymond Brantley wrote:

1 kg/hr is about .0081 CFM. And this is at 20oC (68oF) at 1 atm. (sea level). A mass air meter is that; a MASS air flow meter, not a volume meter. The units make sense [Kg/hr], if you’re an old time, red-blooded American engineer or scientist that would be [slug/hr] where 1 slug = 14.59 Kg.

Here are the conversion factors for CFM to/from KG/HR. They are based on the Ideal Gas Law:

```
   KgHr = InHg * CFM * 36.2 / (TempF  + 460)
   CFM =  KgHr * (TempF + 460) / (InHg * 36.2)
```

InHg is barometric pressure.
TempF is Temp in deg F.

Here's some info for your V/E calculations. This is the info for an 89-93 Mustang V-8 stock MAF meter, mapping voltage to Kg/Hr. If you have an after-market meter calibrated for different injector size you can subtract these percentages from your voltage across the scale. 24 LB 21%, 30lb 37%, 36lb 48%. Or just 19 divided by whatever injector size the meter is calibrated for.

Airflow in Kg/Hr	Meter Voltage
13.307	.57
18.377	.75
23.446	.88
32.318	1.09
36.437	1.18
44.358	1.32
53.863	1.47
62.101	1.57
72.557	1.69
82.062	1.80
89.983	1.89
99.805	1.99
119.766	2.15
141.945	2.32
160.955	2.45
182.184	2.58
201.511	2.69
243.017	2.91
295.296	3.15
347.258	3.35
399.220	3.54
499.341	3.89
595.661	4.18
595.661	4.18
695.149	4.44
835.193	4.77

Table 54: Mustang MAF Transfer Function

If you have a frequency meter you can measure the BAP sensor, but they are different before and after 93.

Testing AFMs

To test a MAF, supply it with +12V and ground. The output will vary from roughly 0.25V to 0.5V at no flow, up to 4.75 to 5.00V at full flow.

John Lloyd <john@anergy.demon.co.uk> sent the following MAF calibration tables

"I calibrated an air meter the other day in the lab... A slight discontinuity between the hi and lo flow masters but it may be of use?

Calibration of air meters with Ford AFM

Vs=5.0
Tamb=19C
19-Mar-97
L/min

Lo meter	volts	Hi meter	volts
0	1.113	200	3.045
25	1.113	250	3.339
30	1.113	300	3.564
40	1.113	350	3.766
50	1.113	400	3.854
60	1.113	450	3.971
70	1.262	500	4.076
80	1.463	550	4.158
90	1.824	600	4.201
100	1.882	650	4.245
120	2.262	200	3.097
140	2.515	400	3.868
160	2.63	200	3.087
180	2.83		
200	3.014		
110	2.106		
160	2.629		
0	1.113		

Table 55: MAF Calibration

Below data came straight off a Ford Calibration of air meters with AFM

Vs=5.00
Tamb=19C

AFM1	Bosch 0 280 200 025	19-Mar-97
AFM2	Ford 86GB12B529-AA with ref 0 280 200 047	29-Apr-97
	From 2.9i V6 using two off	

L/min Lo meter	AFM1	AFM2	Hi meter	AFM1	AFM2
0	1.113	0.2	200	3.045	1.16
25	1.113		250	3.339	
30	1.113	0.25	300	3.564	1.73
40	1.113		350	3.766	
50	1.113	0.25	400	3.854	2.09
60	1.113		450	3.971	
70	1.262	0.25	500	4.076	2.35
80	1.463		550	4.158	
90	1.824	0.25	600	4.201	2.58
100	1.882	0.25	650	4.245	
120	2.262	0.45	680	2.75	

Table 56: MAF Calibration

L/min Lo meter	AFM1	AFM2	Hi meter	AFM1	AFM2
140	2.515	0.68	400	3.868	
160	2.63	0.83	200	3.087	
180	2.83	0.98			
200	3.014	1.15			
110	2.106				
160	2.629				
0	1.113				

Table 56: MAF Calibration

This MAF transfer function graph came from Mike Wesley’s Calibrator demo.

EXTERNAL (SERVICE PORT) CONTROL

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

EEC MEMORY CONTROL

The EEC uses an 8 bit multiplexed address / memory bus (MB7:0) and three control lines for all memory I/O. Unlike the common Intel multiplexing scheme, there are only 8 I/O lines and 3 control lines total, but these are set up with the capability to be used very efficiently.

Unlike conventional RAM and ROM that must have its addresses decoded externally and then be given a "Chip Select" signal, both the EEC RAM (typically an i81C61 24 pin DIP), and the EEC ROM (typically an i8763 24 pin DIP) have their addresses decoded internally. The memory system’s efficiency comes from the fact that there are two internal address registers in the ROM, and three internal address registers in the RAM. The ROM and RAM each have a 16 bit Data Address Register (DAR) that is loaded with the required data’s address in two chunks (8 bits at a time) from the MB bus. The neat trick is that they also have a second register that is a 16 bit counting register called the Slave Program Counter (SPC). The SPC register always mirrors the 8061’s internal Program Counter (PC) register. As long as there are no program jumps, the 8061 only has to issue a command to retrieve the next program byte from memory, and the data appears, it doesn’t waste time sending an address like most typical microprocessors.

Both the i81C61 RAM and the i8763 ROM’s internal address registers (DAR and SPC) are a mirror image of one another. If information is requested from the memory location pointed to by the SPC, the two memory chips check to see if the address is in their memory map. If the address is in one of the chip’s memory map, then the data at that location is output to the processor via the MB bus. The result of this scheme is that if the RAM and ROM are not matched sets, it would be possible to have RAM and ROM trying to drive the MB bus at the same time. This isn’t going to happen when using the original EEC chips, but could cause problems if someone is using an improperly designed memory translator "CHIP" in the EEC.

In order to "reprogram" an EEC, tuners have resorted to building "ROM DECODER BOARDS". What these boards do is translate between the EEC MB Bus and the typical "ROM" chips that are commercially available. These boards (in one form or another) have two 16 bit registers, one of which is a counting register that is used as the SPC, and the other is just a regular 16 bit "Latching" register that is used as the DAR. At any given time one of these two addresses is being fed into the standard ROM that the board contains. In addition to these two registers, there is an address decoding section that allows the ROM’s outputs to be driven ONLY if information is being requested, AND the address falls within the original (EEC’s) ROM’s address. The typical EEC ROM is mapped from memory location 0x2000, so there is also some "address swapping" that allows the standard ROM to begin its memory mapping at what looks like 0x2000 to the EEC.

These "ROM DECODER BOARDS" have only three control lines to work with, the /STB, IT, and DI. These three lines are controlled only by the 8061 microprocessor, and not any of the other chips attached to the MB bus.

- /STB is used to tell the memory that it is time to "LATCH" the 8 bits of information on the MB bus (If the 8061 is driving the bus), or that it is time to put the next byte of data on the MB bus (if the MEMORY is driving the MB Bus).

- IT can be thought of as the "Instruction Transfer" line. What it really does is choose whether the 8061 is accessing the SPC (if IT is high) or the DAR (if IT is low).
- DI can be thought of as the "Direction Indicator" for the MB bus. If DI is high, then memory has the rights to the MB bus. If DI is low then the processor has the rights to the MB Bus.

(DAR - data address registerSPC - slave program counter)

An example would be reading a memory location using the DAR. In a "NORMAL" memory system (one with an address and a data bus) if we were going to read a memory location we would:

1. Place the address on the address bus.
2. Output the memory "read enable" to "enable" the memory chip.
3. Read the information off of the "DATA" bus.

On an EEC MB bus system we would follow the same pattern:

1. Write the DAR (give the memory's address register the address).
   a. Lower DI - this gives the rights to the MB bus to the 8061.
   b. Lower IT - this tells the memory system we're using the DAR, not the SPC.
   c. Drive MB bus with low 8 bits of the address.
   d. Bring /STB low for a cycle - tells memory to latch the low address.
   e. Drive MB bus with high 8 bits of the address.
   f. Bring /STB low for a cycle - tells memory to latch the high address.

At this point the memory system has our "ADDRESS" that we want to read.

7. Raise DI - This tells the memory system that it has rights to the MB bus.

   This is the same as sending a "memory enable" or "memory read" signal.

8. Read the data off of the MB bus.
   a. Read the low byte off the MB bus.
   b. Bring /STB low for a cycle.
   c. Read the high byte off the MB bus.

One note: The EEC uses a "16 bit" memory, so an address didn't have to be supplied for the 2nd 8 bits of data, but when using the DAR you must begin reading from an even address, and the odd address will be available the following /STB. If you use the SPC to read the memory, you can (in theory) just keep strobing and read the entire chip. I've found that I can only "strobe" about 300 bytes of data out of the memory before it "clams up". Your results may vary.

MEMORY CONTROL FUNCTIONS			
DI	IT	/STB	
0	0	1	CPU SENDING ADDRESS TO DAR
0	1	1	CPU SENDING ADDRESS TO SPC
1	0	1	MEMORY SENDING DATA POINTED TO BY DAR TO CPU
1	1	1	MEMORY SENDING DATA POINTED TO BY SPC TO CPU (USUALLY AN INSTRUCTION)
0	0	S	CPU TELLING MEMORY TO "LATCH" DATA INTO DAR
0	1	S	CPU TELLING MEMORY TO "LATCH" DATA INTO SPC (PROGRAM JUMP)
1	0	S	CPU TELLING MEMORY TO "BRING UP" HIGH BYTE <--(DAR + 1)
1	1	S	CPU TELLING MEMORY SPC = SPC + 1, AND PUT ON MB BUS

Table 57: Memory Control Functions

NOTE: "S" indicates a "STROBE" of the /STB line. /STB is active low, and is never low more than a single machine state. There are three machine states per processor "clock" cycle.

Service Port Adaptors

Most commercial add-ons use a 27c512 chip, and the processor on the board only switches to the outside from address E000 to FFFF (which is where the calibration area / maps are for the EEC V).

The Superchips, Hypertech, and Wesley chipmaster ROM boards contain a PLD from Cypress. Small device - not very complex at all, not many gates. Easy to copy as no security provision on the Cypress parts if you have access to the full development kit (you can read it back) and then make more copies.

The PLD on the Calibrator is much larger and uses an Atmel part. It’s very fast and has lots of I/O pins and internal gates. From the size of the part, it’s either a complex job or he’s using a much bigger part just to confuse people. Atmel parts also have a security bit that can be set that makes the part unreadable.

PLD’s

The "simple" PLD devices today usually have what is called a macrocell associated with each output. The macrocell has an AND/OR array that is fed into a flip-flop. The output associated with the macrocell can be programmed to take either the output of the flip-flop or directly from the AND/OR array, so they can be either REGISTERED or COMBINATORIAL. The flip-flops can usually be configured as RS, T-type, D-type, and on some, JK-type flip flops. As the devices get more complex, you can have ’buried’ macrocells that have no external outputs, but can be used to build state machines inside the part.

Since the 8061 is using an extreme example of a multiplexed bus, the PLD has to use registered outputs. Actually, that’s not entirely true. You can build transparent flip-flops from the combinatorial logic because the output of each of the macrocells is fed back into the device and can be used as an input as well. So you could do simple latching without registered outputs, but you also have to implement a counter for the 8061, so registered outputs are necessary. In the case of Wesley’s cal-pak things, it gets a little more complex. He routed the output of the EPROM into the EPLD (or at least he used to), so the pins that are connected to the 8061 address/data bus have to alternate between inputs to the address latches and an output buffer from the EPROM.

EEC-IV / EEC-V FLASH MEMORY

There’s been a guess that to program the EEC-V flash memory requires use of the J3 BS0, BS3, ERASE and PROGRAM pins. To find the FLASH memory, look for a PLCC package, probably 32 pin, that should be a standard 28F010 12V write, erase/5V read Flash available from TI, Intel, AMD, and others. It might be labeled 81C65 or 81C62.

Also, there’s some evidence that there’s a "transitional" EEC-IV about 1993 that used the EEC-V FLASH memory but still retained the EEC-IV connector. This could be a boon to the tuner, as no adapter boards would be needed - only a cable to the PC and the protocol and software to reprogram it.

Investigation is underway to accomplish real ’scan tool’ type data to be retrieved via the STI and STO lines of the EECIV. Apparently, STI goes to an HSI port and STO goes to an HSO port. Currently, there is a piece of code that is added to the interrupt routine that allows sending an address to the EECIV via the STI line and then echoes out the result from the STO line. It needs two hooks into the EEC -- one for a periodic interrupt (for the STO data) and the other for the HSI interrupt.

WARNING! The sensor +5 volts is the SAME as the CPU +5 volts. Any accidental connection of the +5 ref of the TPS sensor to the battery via a test light, and spitzensparken and poppencorken.

The biggest problem I can see when re-calibrating an EECIV is that there is virtually no way of figuring out what it is up to. Perhaps a ’scan port’ patch would be useful? How I think you would need to do that is enable the EER, and modify a few bytes of Ford code to get a hook for ours. Then out of the STO you can get scan tool data like you see on a GM. The limitation here is someone needs to figure out the use of any interesting memory locations and the scaling factors.

This means when you get data, the SES lamp flickers.

I think J3 pin 12 MRESET and 22 TSTSTB are the ones to use with the pause pin to

read the ROM from J3. You have the unit full on and in pause mode by putting a low on J3 pin 8 and then putting a low on pin 12 and releasing (to reset memory program counter I guess) and then keep strobing pin 22 TSTSTB low and reading MB after each strobe and it seems to be reading the ROM contents. It appears STROBE pin 21 is for reading and control from CPU but MSTROBE is for memory dump.

Another contributor wrote:

In my opinion, the whole memory was set up to be high speed, yet only 8 bits wide. A conventional computer has to use the bus to apply an address, and then another bus to read the data at that address. It appears that, in an EEC, the CPU can just request "the next program byte" -- as pointed to by the "Slave Program Counter" so there is no need to send an address. There is a second register, the "Data Address Register", which is in charge of the data. It, however, doesn’t increment by itself. The advantage of having both of these registers is that one can read an instruction, download the address of some data that’s wanted, read the data, and then read the next instruction WITHOUT SENDING AN ADDRESS FOR THAT NEXT INSTRUCTION. I still have to be able to load that Slave PC with another address though, in the case of jumps and interrupts and stuff like that.

I’m guessing that getting the high and low bytes of data is even easier than I’m showing. Why else would they call it a "16K X 16bit ROM"? If you have to ask for the high and low bytes separately it would be more accurate to call it a "32K X 8bit ROM"

Technical Notes on The EEC-IV MCU

EEC EFI MODIFICATIONS & UPGRADES

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

SD TO MAF CONVERSION

Information on MAF conversion sent to me by Bob Nell bnell@utk.edu

(This is specifically for ’87-’88 SD 5.0 Mustangs.)

I started with a stock ’88 Speed Density 5.0 LX and got the MAF sensor, bracket, tubing, EEC-IV and manual tranny from a wrecked ’91 LX MAF 5.0. (The Cobra MAF is different than the GT/LX MAF in that,in the Cobra, both the EEC and the MAF have been recalibrated for 24# rather than 19# injectors. So, if using Cobra components, use a Cobra EEC.) The MAF connector was purchased from a dealer for about $10 and came with solder, shrink wrap and splicing instructions. If a complete harness is required, there are companies that sell used ones.

You will need:

- Mass Air Flow meter calibrated for the injector size (using mismatched parts can cause part throttle detonation and drivability problems).
- MAF bracket and mounting screws (3).
- MAF 4-wire connector, wiring to go to the EEC connector and EEC connector pins for two of the wires.
- MAF ductwork -- the tubing that goes from the engine to the MAF, from the MAF to the air cleaner and then to the intake scoop. Some clamps will also be needed to hold it in place.
- Extra EEC connector pins for hooking up the VSS and FPM2 signals.

The first step will be to remove the old speed density computer from behind the kick panel on the passenger side. Remove the kick panel, the retaining screw on the lower right of the ECM, push the wires out of the way and then, using a 10 mm wrench, remove the ECM.

There are unused female pins (sockets) in the harness end of the EEC connector. Seven of these will be needed to add the necessary inputs (it can be done with four, but it’s not recommended). Looking at the connector, there is a red "H" piece of plastic that needs to be popped out of the connector to allow the pins to be inserted/removed. It can be pushed out from behind (the wire side) then a small pick will be needed to allow the pins to be removed.

To be added:

- 4 Inputs for the MAF.
- 2 Inputs from the VSS (if you don’t already have cruise control, you will have to add a new speedometer cable with VSS and appropriate wiring).
- 1 Input from the Fuel Pump Relay.

If cruise control is present, run the 2 VSS inputs to the speed control amplifier instead of routing wires outside the car to the tranny. The amplifier is the yellow-ish plastic box located near the driver’s left foot. There are also a couple of pinout changes between the speed density and the MAF EEC that require swapping a couple of pin positions - if it’s as remembered, they are the TAB/TAD (thermactor air bypass/diverter) pins.

Attach these 4 wires from the MAF to the EEC

- Air Meter Pin C-T/LB to EEC pin #9
- Air Meter Pin D-DB/O to EEC pin #50
- Air Meter Pin A- Red to EEC (splice into the existing red wire on pin #37) (this is VPWR)
- Air Meter Pin B- Black to EEC (splice this into the existing blk wire on #40 or #60) (this is PWR GRND)

Also, these changes must be made:

- Pin 51 must be moved to pin 38 on EEC
- Pin 11 must be moved to pin 32 on EEC

To hook up the VSS:

- VSS + must be hooked up to Pin #3 on EEC
- VSS - must be hooked up to pin #6 on EEC

The VSS signal can be taken from the VSS or it can be tapped off the speed control amplifier, which is located near the dead pedal, and is the yellowish box in the corner by the driver’s left foot. The DG/W wire is VSS+ and the black wire is VSS -

To hook up the Fuel Pump Signal:

Splice into the PK/BL wire that goes into the Fuel Pump Relay (located under the driver’s seat on pre-93 Mustangs) and run it to pin #19 on the EEC.

I don't know if anybody will use these, but it's nice to have the pt#'s

Motorsport Harness's and Component Kits

Main Harness M-12071-C302

Engine Harness and Controls Package M-12071-E302 5.0/ HO & 5.8L

Firing Order 1-3-7-2-6-5-4-8

Engine Harness and Controls Package M-12701-H302 5.0L Non HO

Firing Order 1-5-4-2-6-3-7-8

Engine Harness & Controls Package Includes: Injector Harness & Intake

Mounted Sensor Package(ACT,ECT,EGR Pos. Sens., EGR), along with O2 sens.

harness

Sensor and Relay Package M-12071-D302

Includes (2) O2 sensors, BAP sensor, EGR, TIB, and TAB solenoids. Fuel, air conditioning, and EEC power relays.

Computer and Mass Air Meter Kit M-12071-F302 (Man. Trans.)

M-12071-G302 (Auto.

Trans.)

Includes MAF Meter, EEC-IV computer, MAF inlet and outlet hoses, and MAF meter mounting bracket

NOTES FOR TRUCK APPLICATIONS:

The ’95 Mustang Cobra R, (351W) used an 80 mm Lincoln Mark VIII MAF and 24# per hour injectors. The injectors will easily support 350 HP and the 80mm MAF is a better choice than the 70mm, as you get to use more of its linear range, so fueling can be more accurate.

To convert SD trucks with E4OD/AODE transmissions to MAF, some people use the CA 5.8 MAF/E4OD (F5TF-12A650-BYA). Mike Wesley says he uses the F5TF-12A650-HB (95 CA 5.0 MAF/E4OD) on a 750+ HP daily driver 415 stroker Lightning with a Vortech S trim. It runs open loop, has been reprogrammed, drives like stock, gets 17 MPG and will run low 10’s at 130+ in the 1/4 mile and A/C and cruise work great. Both of these EEC’s are set to use 4.10 gears. If a smaller ratio is used, say 3.55, you could use the F5TF-12A650-GB. There are probably 15-20 EEC’s available to convert a SD (later model) to MAF (some of these may compromise the ability to stay emissions legal).

If you have an early SD truck with AOD, re-wire to the Mustang EEC (Ford MotorSport sells this kit). You’ll have to move/add quite a few wires, and you might not like the results if you’re not able to re-calibrate the EEC (like the Pro-M ’low cost’ kit, Kenne Bell, LCA and Downs Ford come pre-re-calibrated). The engine shuts down at 85 MPH, shifting is fairly sloppy and too early (at least on a Lightning). All Ford EECs shift poorly -- except for the Lightning, which is only slightly firmer.

To use the Mustang EEC on a truck with an E4OD/AODE, you would need to run two EECs in parallel. The Mustang EEC runs the engine, the existing truck EEC controls the trans. Pro-M sells a kit like this.

As with any instrumentation and control system, poor or improper grounding of sensors can should be avoided. The TFI module seems to be the most susceptible to problems -- and especially when making a new or modified installation. The largest source of current-induced voltages in the grounding system is the starter current, with the alternator being the next, so every care should be taken to isolate the engine management circuit grounds from the starter current path.

Noise generated by the alternator and by aftermarket CDI (including multi-spark units) ignitions can cause real grief with an EFI. This noise can be both radiated and developed across shared grounds so, in addition to proper grounding techniques, care must be taken to not run wires close to the ignition/coil wires and to provide shielding where necessary. Some recommend placing a capacitor at the supply for the CDI to absorb the transients it generates. This is a good idea, but keep in mind that, when capacitors are involved, larger is not necessarily better. A large, aluminum electrolytic capacitor will begin to become inductive (i.e. it stops working as a capacitor) sometimes at frequencies as low as 10 kHz. In actuality, one would be better off with 100 µF capacitor with a .1 µF disc ceramic in parallel with it. The larger capacitor is adequate to provide the long “drinks” the CDI pulls from the battery voltage, while the smaller disc ceramic works up to hundreds of MHz and will help suppress the very fast rise time voltage spikes that are the true source of the radiated noise.

Also, keep in mind that the EEC stops working when the input voltage drops below about 8 volts -- another reason to keep a good battery and to not share power distribution paths with the starter.

The starter and charging circuit current loop should not include the engine management ground current loop, and vice-versa. As much as possible, the ground current path for the alternator and the starter should be kept entirely separate from the ground current paths for everything else. The best arrangement would be a heavy (e.g. 2 AWG) ground wire from the battery negative to the engine block at or near the starter and a similarly large wire from the battery positive to the starter relay. The +12 volt power for the ECM and the accessories can be picked off at either the starter relay or directly from the battery terminal. The ECM/accessory ground should be taken directly from the battery negative.

Note that steel and iron are poor conductors and that much of the ECM sensing circuitry uses the engine or chassis for grounds. To head off problems before they start, heavy, stranded copper wire ground straps are recommended between any component that has a sensor bolted to it (e.g. the intake or the block) and to the ground at the battery. Solid copper tubing can be run along the chassis and/or frame and bolted to it at several points to assure a good ground to even the remote fuel pump and sensor -- just remember that stranded welding cable should be used anywhere the conductor can vibrate, move or flex so it won’t break over time. (If you’re not sure about why this is necessary, measure the resistance from some sensor’s grounded case back to the battery ground and keep in mind that the voltage dropped across that same path is E = I * R. So, a resistance of only .1 ohm will have one volt induced across it if 10 amps were to travel through that path, and the resistances are often much higher than that.)

TUNING

(by Bruce Plecan with editing by Joe Boucher and then by me)

Introduction

This purpose of this document is to assist someone in the tuning stage of an Electronic Fuel Injection (EFI) project. It contains discussions about the monitoring equipment required and the methods and strategies for the actual tuning. The information is based on the experiences of tuning a GM EFI setup on a small block. However, the methods should be applicable to most any engine.

This paper doesn’t reveal the location of the maps and tables in the ecm software but it does discuss their functions. The maps and tables for some ecm’s can be found on the internet. For example, the Syclone/Typhoon and the Buick T-Type fanatics have found most, if not all, the needed information to alter the EFI systems for the extreme modifications on their cars.

Look out, here comes a warning. You can really mess up an engine if you don’t understand what is going on and are not careful and methodical about tuning your EFI setup. This is not lawyer gibberish, this is for real. You can turn that expensive, close tolerance machine into a pool of molten crap real quick. I personally know of a kid who had too much money and not enough smarts. In the early days of EFI, he bought an O2 sensor monitor and a device to alter the O2 feed back signal to the ECM. He didn’t hook up the O2 monitor properly, started twisting the knob on the signal modifier, leaned the fuel mixture too much, and melted the pistons. It can be done! After all the time and effort you have put into achieving this stage, it would be a shame to ruin something because your enthusiasm carried you away and you tried to tune too fast.

What Is Being Modified

When tuning the EFI setup, the data controlling the fuel mixture and spark timing is being altered. Each engine type and setup will require a unique set of data to achieve its ultimate power and efficiency setup. The camshaft, intake manifold, head design, and exhaust system all influence the torque curve of the engine and therefore the fuel requirements. The spark advance plays into this also. The following section discusses the different tables found in some ecm’s.

List Of Tables

Tables, Maps, Switches seen or heard of: Main Fuel (VE or LV8); Fuel Enrichment WOT; Main Spark (rpm vs. kPa or gms/sec); Spark added WOT Idle Speed; IAC steps (by cool temp); WOT enable TPS %; TCC lock/unlock by road speed; TCC disable by TPS %; TPS Enrichment (sensitivity, quantity, duration); Injector constant; AFR % enrichment vs. Coolant temp; EGR temp enable; EGR enable % TPS; EGR timing correction table; EGR fuel correction table; Speed limited MPH; RPM Limiter; Degrees retard by knock sensor by MAP; Degrees retard by knock by rpm (WOT); Turbo Boost Control; All diagnostic malfunction disables. Not all ECMs use all these tables!!

DISCUSSION OF TABLES

The following tables are for a 3020 lb. car:

Engine; AL headed (vette 58cc chamber), 355 CID SBC, 9.5:1 CR
Cross fire intake; 2" butterflies, .25" lid spacer 90lb/hr injectors
Cam specs; 270 Comp Cam, .470 lift 270 adv dur, 224 @.050 110 3.42 gears
Tranny; 700R4, 2800 rpm non-lock-up stall
Exhaust; Edelbrock TES manifolds, very low restriction cat, 3"single ex (Walker).
ECM ; GM 1227747 (no internal mods, OEM v-8 netres [a resistor pack])
Some details are not mentioned (gotta keep an edge).

	30	**35**	**40**	**45**	**50**	**55**	**60**	**65**	**70**	**75**	**80**	**85**	**90**	**95**	**100**
400	0	0	0	0	0	12	16	16	16	12	8	8	8	8	8
600	0	0	0	0	0	12	16	16	16	12	8	8	8	8	8
800	-4	-4	-4	-4	-2	12	16	16	16	12	8	6	4	4	4
1000	-6	-6	-6	-4	0	12	12	12	11	8	6	4	4	4	4
1200	-6	-6	-6	8	10	14	14	12	11	8	6	4	4	4	4
1400	-6	-6	18	20	20	18	14	12	11	8	6	4	4	4	4
1600	-6	-6	24	24	20	18	14	12	10	8	8	8	6	6	6
1800	-6	-1	27	28	24	22	17	12	10	8	8	8	8	6	6
2000	-6	-1	30	30	26	22	16	12	12	10	8	8	8	8	6
2200	-6	-1	30	30	26	22	18	16	14	12	10	11	8	8	6
2400	-6	-1	30	30	26	22	18	16	14	12	12	12	10	10	10
2800	0	0	26	26	26	22	22	18	18	17	16	14	14	14	14
3200	0	0	24	24	24	22	22	20	18	18	18	18	16	17	17
3600	0	0	22	22	22	22	20	20	20	18	18	18	16	16	16

Table 58: Spark Table (RPM vs Vacuum kPa)

RPM	800	1200	1600	2000	2400	2800	3200	3600
Advance	0	0	0	1	6	6	6	6

Table 59: Power Enrich Spark (RPM vs. Added Spark Advance)

	20	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
400	0	0	0	16	16	36	41	50	54
800	0	0	0	16	18	39	47	52	54
1200	0	0	0	29	45	50	52	57	60
1600	0	0	25	46	50	51	54	58	60
2000	0	0	51	49	51	53	57	59	62
2400	0	1	49	49	51	54	56	59	62
2800	0	2	50	52	54	55	57	60	64
3200	0	3	50	53	55	57	59	61	64

Table 60: Volumetric Efficiency (MAP vs. RPM)

RPM	0	400	800	1200	1600	2000	2400	2800	3200	3600	4000	4400	4800	5200	5600	6000
Add Fuel	10	10	41	40	41	41	39	37	35	35	37	39	41	41	31	16

Table 61: Volumetric Efficiency WOT Enable (RPM vs. Additional fuel factor)

RPM	0	400	800	1200	1600	2000	2400	2800	3200
% TPS	35	27	26	25	24	23	22	21	20

Table 62: %Throttle Position Sensor for Wide Open Throttle (RPM vs. % factor)

These tables are very different from the OEM setup. But, for this engine combination, it works. The items to note are the lean/retard overrun (high vacuum values when you lift your foot off the accelerator pedals at speed) areas. Glance at the spark table to see when the engine wakes up. Note that the VE table values can add up to over 100%. Also, because the values in the %TPS for full throttle table are low, and with the existing VE in the first table, the system will change the air fuel ratio to rich instantly. Might also note the funny timing under 2800 rpm. Remember the 2800 rpm stall converter. With this combo the engine can light em at will in 1st, and 2nd. The car runs in the 13’s but you can see how mild the cam and engine combination is.

Tuning Tools

In order to know if your wonder toy is about to melt down on one end of the air fuel ratio spectrum or put a smile on an Arab Sheik’s face on the other end of the spectrum, you need to monitor the fuel injection system and some of it’s sensors. Here’s a list of tools:

* Digital Volt Meter (DVM) - (10 meg ohm impedance) for monitoring the Throttle Position Sensor

* Vacuum Gage - For idle and part throttle analysis;

* Tachometer - If you don’t know, you’ve got a lot of learning to do;

* Performance analyzer - To give you an unbiased reference of the car’s performance changes other than the uncalibrated seat of your pants;

* Scan Tool - To know what values the ecm is using internally;

* Oxygen Sensor Monitor - Usually a 10 element LED Display.

(The 10 LED AFR Display, and Knock detector schematics are at the GN T-type tech area.)

O_2 Sensors And Indicators

As clarification, this O2 discussion is about/for conventional V-8s, Small Block Chevy, Small Block Ford, Big Block Chevy, Big Block Ford. While there are probably numerous others, there are a bunch of VTEC types that this may not be accurate for. Monitoring this sensor will keep you from melting your engine.

From some personal notes from the early-mid 80's there were many 2 gas analyzers that went to market for the then pending inspections for CA. Here are some conclusions drawn from some GM training center notes and a Radio Shack engine book. First from a slip of paper, circa '83, that lists O2 sensor volts relative to AFR Air Fuel Ratio, and one from 2-3 years ago from Cyberdyne, who manufacturers a 10 led AFR Monitor. The voltages match i.e.: 1v=17:1; .3v=16; .5v=14.9 ; .7v=13.8:1; .9v=12.7:1. Also, going by the RS book are some differences in response time vs. temp, and differences in sensor voltage depending on load. So yes there are variables, and yes the results will vary with mounting locations, and loading. But for high performance tuning as an INDICATOR, there are no reservations on using one as a guide. The double check would be doing plug cuts. If you are working on a 20:1 AFR engine, then you would have to use a wide ratio O2 sensor, or if it was exposed to a limited amount of exhaust gas and needed heating and thermal compensation then you need the double wammy wide ratio O2 stuff. For what the average DIY will be attempting, a good ole GM single wire mounted in iron manifold, and a 3 wire used in tubular headers will work just fine. So while the ME's, and EE's discuss the merits of both for a lot of us just building the 10 LED O2 monitor like is shown in the GN T-type tech area will work fine (http://ni.umd.edu/gnttype/www/buick.html). Also, the GN's have a bunch of cars running 10s,11s, that regularly use unleaded with a heated O2 and have "fair results" (10s with A/C actually is better than fair in my book).

Where To Start

Hopefully, when starting with any custom ecm or drive train combination, you'll have something to start with, and better yet at least an EPROM from the same engine series. If not, then we'll cover both anyway, again with the GM OEM, because that's what was experimented with.

Regardless of what you're working on, you have to pick a spot to start with. This involves pulling numbers out of the hat to begin with. The more you do, the smaller the hat gets, because you can rule out some things.

Idle

A suggestion is start at idle. For a SBC the initial timing for the distributor is 6 degrees before BTDC. How do you setup 6d without a timing light? On GM style distributor there is a set of little arrows inside the distributor. On a V-8, 8 on the reluctor, and 8 on the sensor ring. Just line the marks up and tighten the distributor clamp.

The engine fires when the arrows are ever so slightly past each other in the direction of rotation. The second you get to about 400rpm is when the engine starts running off your ecm control. Less than that and on a cold day the engine will windmill when it's cranking. Fill in the 400-800 RPM areas of the timing map with

16d (Table 1). Also, fill in the lower rpm VE tables (Table 3) with 30, and try to start the engine without grinding the starter too much. The engine will warm up enough to wake up the O2 sensor, look at the scanner, and see where the BL is, and IAC Counts. The IAC Counts are a vacuum leak. So they are critical to a decent idle.

Now you have some idea of where the mixture is, and what the kPa is from reading the scan tool. From experience you can't run the same numbers everywhere at idle. On the high side of idle kPa go richer, i.e.; with an engine that idles at 60 kPa if it is 30 there, at 65-70 it would be 32, and at 50-55 28. Yes, the 30 is a lot but its easy to go lean, and clean up the idle, then having stall all the time, and having to wait for the prime shot to burn out.

Next, lean the VE values (Table 3) down till the O2 is swinging (value moving back and forth through .45 volts), the IAC is less than 10, and you have a min idle speed. For the initial butterfly opening I cut a strip of paper 1/4" wide and set the throttle idle setting so the paper is pulls through the butterfly opening with a minimum of effort and without tearing. Now the engine starts and will idle. This is a good start.

Light Cruise

This section is on how to do light throttle cruising set ups, using GM ECMs. This is for an engine using a slightly hotter than stock cam, healthy ignition and assuming that some roughing out of the fuel timing tables have already been done.

The objectives are: Advance timing as much as possible consistent with minimum throttle opening; and use just enough fuel to run 128 or slightly lower BL numbers.

For monitoring the engine, a digital volt meter (DVM) is attached to the Throttle Position Sensor (TPS), a vacuum gauge is hooked up to the manifold (but this is almost redundant of the TPS), a scan tool is hooked up for looking at the Block Learn (BL) numbers, and a 10 element LED display is attached to the oxygen sensor. With this wired up it is easy to now start changing the timing of the 40 kPa areas of the fuel map by rpm for cruise. On a flat road with the car in top gear, find the cruise speed at 40 kPa and note the DVM signal from the TPS. Add timing and repeat the road test at the same TPS value. Observe whether the vacuum value is above or below the original signal. Repeating this procedure will reach a limit where the knock sensor will start retarding timing. Next, drop off as much timing going to the 20 side (overrun), which richens the O2 readings and helps to cut the fuel on overrun, so the car slows when you take your foot off the gas. On some of the OEMs you almost have to hit the brakes to slow the car.

From the 40-50 kPa vacuum value, again depending on your car, drop off as little timing going toward 100 kPa (full throttle), as possible. Run enough fuel in the VE table so that when pulling slight grades the motor doesn't chug (love that technical language). Also use a minimum amount of timing for the WOT spark value.

A suggestion is stay on the low side of the BL 128 at cruise so the engine is instantly smooth and pulling when the gas is nudged. There is no difference in MPG between this and 128. If you have the capability for pulse width modulation (PWM) of the EGR I'd turn this off during the initial tinkering so that you don't chase too many items at one time. Just doing the spark and fuel can be an exercise in patience. Also, with a big cam just off idle and just as you go to light cruise at say 30-35 mph, you may have a dodo area (in this area of the PROM maps, BLs are lousy regardless of timing AFR). What is happening is the engine is starting to wake up, the intake tract length is starting to work, and the engine is reducing its self EGR. Pulling from 1600 rpm in high gear, smoothly from a roll might be the best performance test possible. Some engines just don't like to lug down to 1400.

Perform the light throttle cruise initial tuning in whatever gear is 1:1, or less (overdrive). Doing the tuning while giving the engine the advantage of gearing might leave you having to do a major retune.

On some of the EPROMs you can lower the TPS enable settings. If you're running an auto tranny watch it. If you have a throttle value (TV) cable sensing transmission you must have enough line pressure increase to keep from smoking the tranny to handle the extra HP at low TPS settings.

WOT

General Concepts

Remember, this is from personal experience, and the example vehicle is the vehicle used in the light cruise tuning. DON'T DO THIS WITHOUT ADULT SUPERVISION. In other words, be careful, patient and methodical. To review, starting with a GM ECM with initial programming set to 10-14 degrees WOT maximum retard. Set the decay rates slow. Set the total timing amounts to 30 degrees. Set the WOT AFR to 11.5. Hook up a scan tool, LED O2 sensor display, and performance analyzer.

Now adjust the F/P so that you can drive at WOT. Watch the O2, as you drive (OK glance at it). After the pass is over, look at the knock counts. If the O2 sensor was plenty rich (all ten LEDs Lit), and no knock counts advance all timing 2 degrees and lean the WOT A/F mixture to 11.7

Repeat driving test.

A few words about the last item, analyzing performance requires something self-triggering, and easy to use. A home-grown performance analyzer is described in Appendix A.

Now adjust the F/P so that you can drive at WOT. Watch the O2, as you drive (OK glance at it). When the pass is over, look at the knock counts. If plenty rich (all ten LEDs Lit), and no knock counts, advance all timing step 2 degrees and lean to 11.7

Repeat driving test.

Plenty of LED, and no knock counts, add 2 degrees more timing and lean the mixture to 11.9.

Somewhere now you'll start either slowing down, or tripping the knock sensor. Usually the knock starts triggering between 2000-2600 rpm.

The WOT AFR stored in the prom can be a wide range of values depending on your injector sizing, and PSI. Remember this is an ECM calculation and has nothing to do with reality.

During all of this do some plug cuts, and learn how to read the removed plugs. If the F/P is changed then the fuel VE values at part throttle in the prom tables will have to be adjusted. Some juggling back and forth between the values in the VE table and the WOT AFR will be expected.

Now, the ecm has been told a huge lie. This means we didn't know where the injector constant lives. The TPS Enrichment (EFI talk for carb accelerator pump) location was unknown, so for this application huge injectors were used. The engine is running a 90#/hr injector, with an injector constant in the EPROM size of 65. This is not the best way to do this. But, by creative use of Fuel Pressure Regulators, and time consuming recalibration of the VE tables, the system was made to work, but some of the constants look really strange, i.e. WOT AFR of 13.5. The actual AFR is closer to 12.5 -- the 13.5 only exists in the ECM's mind.

To achieve the EPROM files for this setup, burning 400 chips was not required to get it right, 400 chips were burned to get it PERFECT. More than likely 98% of the world would never notice the difference in the last 100+, and the first couple of 100 was figuring out how to do the above.

PS When doing the above settle for the least amount of timing consistent with the best performance. Trying to run maximum timing results in hitting the knock sensor and having varying ETs. IF you run too much timing you may have to disable the knock sensor diagnostics because the car is so happy with yesterdays tuning, and today the ECM is using the OEM specs and can’t run the diagnostics, and the fault codes get pinged. (This setup is that way)

Mild Performance Engines, Engine Swaps

Now things take a step up, tools are still the same, but now it would be nice to have a scan tool, With probably a different cam the operating range of the sweet spot of the engine has changed, or the weight of the vehicle has. This weight thing is not to be taken lightly (HOHO HAHA), just had to sneak a little humor in. A light car and/or one with higher numerically final drive gears will want a lot more fuel earlier, or more timing taken out sooner. So now you’ll have to have some control of the timing. But still if you’re using a distributor, you can get by with just timing at cruise or WOT.

If it’s a mild cam (and if you’re doing this for the first time I would highly suggest you stay conservative).

Now use your scan tool and look for the magic 128, try adding/subtracting two to four degrees here and there do the 4d to get a feel for things, and do not use any WOT testing when going 4 at a time.

Real Fire Breathing Hot Rods

(Things that go lump, lump, rump, rump at idle, have non-lock up converters, and spin tires in any gear less than overdrive.) This will require modifying such parameters as Main Fuel Table; WOT enrichments; Main Spark Table; WOT Spark Adders; TPS Enrichments (sensitivity, duration and quantity); Enable WOT TPS percentage settings; Cooling fan operation; and Idle speed.

Having the right tools is essential here or the engine will do a basic China Syndrome Routine meltdown -- and it doesn’t take long at WOT. A code scanner, a DVM, vacuum gauge, full time fuel pressure gauge (and isolator), and an air-fuel ratio meter (a DPM or an LED bargraph) are essential and it wouldn’t hurt to have an audible knock detector.

Ok higher compression, no big shakes Small chamber/AL Heads generally make a difference, for either less total timing, at WOT

Aluminum Heads need more timing during light throttle cruise, non-lockup converter, more initial enrichment (due to less drivetrain inertia).

There are several other considerations for the non-lockup converter - including the extra wear-and-tear on the TCC linings. Since more fuel will be needed earlier, the % TPS WOT enable must be lowered.

At this level, the instrumentation requires constant attention to prevent damage. With EFI and knock protection, there is usually protection from things going wrong but, when things do go wrong, you can’t forecast what the problem will be.

Also, at this level, you might consider running a fuel separator, which is basically a large tank for the sediment to separate in, so there is no pressure drop due to a restricted fuel filter.

General Tuning Tips

(Abbreviations: WOT, wide open throttle; DVM, Digital Volt Meter; AFR, air fuel ratio; FP, fuel pressure; VE LV8 mixture; TPS, throttle position sensor, TCC; torque converter clutch)

No matter what, save your original data!!.. Tinker with STOCK ENGINES, to get your feet wet.

One of the easiest errors is to use too much air filter. Actually, the problem is getting the air to organize to flow through the throttle body. Having several hundred square inches of filtering surface area is fine, but having a huge volume of air sitting on top of the butterflies, can have its down sides. I.E. when the throttles snap open, it takes/can take a huge accelerator pump shoot to cover the delay in this column of air to get moving. If you’re using an open element filter then try this. Assemble the air cleaner housing, and element as it would sit on the air horn. Measure the distance from the sealing ring portion to the lid, anything more than an inch is too much. Go to the hobby shop (say you had 3 1/4" dimension) Lets also say the air filter sealing ring was 5" in diameter. You’ll need enough thin brass to make two rings, 3" wide and about 16" long to make a tube 5" in diameter, and 2 1/2" tall. The reason I say two is you have to experiment on the height, and that means going too far, which will ruin the first one. I haven’t found getting more than 1/4" close as necessary.

To get involved all you need is the locations of the WOT enrichments, or the VE tables (TBI), or LV8 (TPI) in the EPROM tables.

With the WOT enrichment/APR you can tune the FP till the cruise feels right, and then enrich WOT. If that doesn’t work try leaning things down. Always remember, too rich and you buy spark plugs. Too lean and you buy head gaskets, exhaust valves, rings etc.

Or tune the WOT, experimenting with the tables to even things out.

Tuning cruise is just watching the DVM for the lowest TPS.

On the Fuel tables changing things by 10% should make a noticeable difference. Again move just a couple at a time, till you get a feel for things.

If you’ve only been driving high performance cars for a couple 100K miles, it might take some getting used to for what makes a change.

If you’ve got a ton of time behind the wheel, and a lot of SCCA time, you might have a better sense of things, either way take your time.

Sometimes a FP adjustment can make the engine respond perfectly. If the fuel pressure is increased, and the engine runs better, that means one thing, it was universally lean. Eventually you’ll reach a point when changing it doesn’t seem to make much of a difference, and then performance falls off quickly. When addition fuel is added (more than what is needed) the ecm just leans things down, so the BL changes. When the engine goes to WOT enrichment it glances over at the BL and if it’s low (less than 128), then it will reduce the injector open time. If the goal is to richen WOT then you have to lean the tables down so the BL is right, or change the WOT AFR. Conversely, to lean out the WOT raise the VE table entries, or lean down WOT AFR.

I have not read anything that says you can’t run vacuum referenced fuel pressure regulators with TBIs, but the pressure output variance is usually too much. But, if you run a second fuel pressure regulator, you can trim this variance. This is the only way out of not being able to tune the accelerator pump features of TBI, since no commercial program I’ve heard of allows this to be done. However, once you have things rather close you can try other chips from other applications, and put your tables on them. From the good old days, a big block always took more accelerator pump than a small block (hint).

Again rich you buy spark plugs, lean it’s internal engine parts. For Spark, not enough timing, and you foul spark plugs, too much advance it’s internal engine parts. (too long, too retarded, overheating)

If TPS enrichments aren’t changeable then try setting the VE high and low, (to try having different BLs), then this BL will start to work as an enrichment during TPS transitions. The reason this works is that the ECM will start moving the IAC counts around.

When tuning remember the IAC is a vacuum leak. Less is better.

A Note About Timing

I have done a bunch of playing with timing especially at WOT, and lately have noticed that 2d less than triggering the knock sensor has had no effect on performance. In the past I’d spent lots of time doing dyno work and many times the customer would wind up taking 2-4d out from where we had it. Please note this is in the early 70’s, and we were using one we made, and were no where as sophisticated as they are now. Might mention this was also getting the AFR to correct within .2 volts at each timing spec.

Try adding timing to the spark table, and remove a similar amount from the WOT spark addition. The same with the fuel table, and WOT enrichment. When you start getting close, change only one thing at a time, one thing only. Move two and things might seem fine but then 2 moves later everything takes a turn for the worse and you won’t know where to start.

Maf System Differences

On a 89 vette there aren’t really the VE tables as shown before, my example was for a TBI which is a Speed Density. The computer knows from various sensors what is going on but has to calculate the air entering the engine, and then fine tunes it by using the O2 sensor. So you have to "give it a range" to estimate where to start these calculations, and based on that a set of look ahead tables for it to consider. Now, on an engine with a Mass Air Flow sensor the ecm is constantly being told how much air is being used, and with that known it’s a simple calculation for figuring the injector timing. So all it really needs is something for enrichment for acceleration, and something for WOT. Setting up a MAF is really easy to do compared to a MAP, if ya can get the WOT enrichments then you about have it all. Some tinkering on the timing might be nice.

The listed items are so tightly wound together that I really don’t know how to separate them for this document. And any one that has more to say on the subject I would invite to comment further. Again this is from my experiences on the subject matter, and my interruption of what others have told me, to correct difficulties that I’ve experienced.

The ecm is a marvelous thing, but it has no self generating logic, it just follows instructions. So it can be lied to and will never know the difference, and generates signals we humans use as a language to understand what the computer is doing. In other words if the computer is told to generate a WOT AFR of 13.5:1 it will do this using the information that is feed it. So this 13.5 is a calculation, and may or may not be an actual real world reading.

On a multiport the injectors are calibrated at 3 bar (bar = 1 atmosphere = 14.7 PSI). Atmospheric pressure is 14.7 PSI. Absolute vacuum is 30 inches of mercury. A typical engine develops 15" of vacuum at idle referenced to a fuel pressure regulator referenced to the engine vacuum signal will change the FP approximately 7 PSI (30 in hg = 14.7 PSI). So the FP changes about 16% (7 psi/43psi) due to this vacuum ref. Depending on its initial flow characteristics the pressure regulator control can change some when it is vacuum referenced when installed in the car. Hence for calibration purposes the ecu value can be set for an injector size of 39 for an actual installed injector that is rated at 42. The less linear an injector is the greater this "error" will be. Also consider rising rate fuel pressure regulators, they will change the FP by a rate greater than the referenced change. I.E. on a turbo motor that generates 14 PSI boost a raising rate could generate a change of 28.

This will help explain why, when the injector size is increased by 10%, you may/may not wind up changing the base injector width by 10% across the VE tables. This would be a good start, then again tune as necessary. It is your decision to start your WOT ranges at 13.5 or 14.7. Another tuner has had good experience using 14.7.

If there is no change possible to these settings, than changing the injector constant will have a very profound effect.

To quote from the OTC (Ottawa Tool Company) Scanner Manual "LV8, Filtered Load Variable (LV8)...in the filtered load variable mode, a number will be displayed that is proportional to the measurement of the amount of air being displaced per cylinder. The filtered load variable replaces the MAP sensor in some applications." So a MAF reading is grams/second, and a LV8 is a number that is a proportion. Therefore in a MAP set of calibration tables you have VE tables, and in a MAF you have LV8. Both have a table of TPS %s vs. RPM to enable WOT. Both have a correction for WOT enable, either a VE adder or % increase in injector timing.

Attachment A

There are some nice $100-$150 performance analyzer units out there, and they don’t have to be perfect, just reliable. What I did was use the VSS and a Radio Shack Stop Watch.

The VSS was tapped at the ECM and that was feed to a LM339.

The stopwatch was taken apart and wires added for the power supply and across the start and stop buttons. The power wires went to an LED that was the ON indicator. The LED voltage drop of 1.5 was enough to run the watch even though it originally had a 1.6v battery (use a 100?fd at the input). The timer runs off a 7805 volt reg. Wire the on + off wires to a 4066. Wire each side of a 556 to be a monostable ON/OFF pulse for the 4066.

Going back to the 339. Tie its output to a counter set to divide by six. There are 3 pulses per tire revolution, so this will take two tire revolutions to start the timer, which helps reduce false triggering from wheel spin. At the reset connection tie that to a second divide by 10 and use that to trigger the first of the 556 for the start 4066. Wire the second divide by 10 to a third one, so that you have a total of divide by 100, and that will be the stop 4066/556 pulse. So now you have a timer that ignores the first 2 revolutions of the rear tire and then counts 200 revolutions to stop the timer figure 6’ rolling diameter ya have a 600’ portable drag strip in car that eliminates error due to wheel spin. (There is something wrong in the math, but paced out it was a little over 1/8 mile. That was 7 years ago.) Could be the last divide by 10 was actually set at 8. Breadboard a 555 pulse generator and burn it in. The watch should run to .001 ever time. No I don’t have a schematic, and it was a cobble together, but it has been working for 7 years, and I not cutting it up to reverse engineer it till it fails. Oh I did something for a timer reset and register clear, but I forget.

EPEC

The following information about SVO's add-on processor that augments the eec came from Super Ford's web site:

http://www.d-p-g.com/superford/april_97/eZ.htm

EPEC consists of the EPEC processor module, a 6 ft. EPEC-to-PC serial interface cable, an EEC-IV module to EPEC module cable called a Wiring Harness Adapter, a short harness to connect to the Input/Output devices, several floppy disks containing the Windows calibration software and a small manual mainly for start-up use. $1395 and a lap-top or other PC is all that's required.

Sam Guido at General Systems Research is the architect of EPEC, which is often referred to as the "Super Extender" by SVO insiders. It purportedly gives everyone the power to make the Mustang EEC-IV engine management do what you want.

For the first time, fuel and spark functions are completely under the control of the operator. It is also a sophisticated data logger, so examining what's happening inside your EEC-IV system, like air temperature, air mass, rpm, etc., is now possible with great accuracy.

SVO is pretty sure the EPEC module will work with any EEC-IV controlled engine, though it's been tried only on the Mustang. Other applications may require that one modify the interconnecting harness supplied with the EPEC. Also, other differences in strategies among the various versions of EEC-IV, like EGR operation for example, may require one to toggle on or off certain functions at certain times, but what and when is something you'll have to discover on your own.

The EPEC is easy to install, thanks to the dedicated wiring harness, and should be operational as fast as you can access your EEC-IV computer and plug in the harnesses -- say 10 minutes. More time consuming will be wiring the seven input/output devices to the short EPEC harness. These are devices you select, such as nitrous triggers and timers, along with data logging hardware such as MAP sensors, thermocouples, etc. The EPEC instruction manual is not in a printed format but is delivered on floppy disks. A short, printed, manual of start-up instructions is packaged with each kit.

EPEC's Windows software offers simple operation. Both table and graph information is plotted.

EPEC works by turning off certain EEC-IV functions and substituting its own. This means EEC-IV's adaptive programming is taken off line, so the EEC isn't constantly battling the EPEC over engine control. Its ability to switch off most of EEC means it has far too much latitude with fuel and spark to remain even remotely within emissions guidelines so it will never be CARB exempted.

For example, to work with the fuel system, you start with the Fuel heading, click and hold the mouse to reveal a menu, select from that menu and flip through the layers of fuel control. Highlight the function you want by clicking the mouse key, enter the parameter you desire and that's about it. Fueling is not handled by spelling out a specific injector pulse width, for example. Instead, there are 11 fuel control parameters available, each with a percentage scale you can easily dial in. The fuel control parameters are:

· Idle A/F Ratio: Controls closed throttle A/F ratio based on rpm.

· Part Throttle A/F Ratio: Controls part throttle A/F ratio based on rpm.

· Wide Open Throttle (WOT) A/F Ratio: Controls WOT A/F ratio based on rpm.

· Air Charge Temperature (ACT) Multiplier: Richens or leans the fuel based on inlet air temperature.

· Engine Coolant Temperature (ECT) Multiplier: Richens or leans the fuel based on engine coolant temperature. In other words, it’s the choke.

· Throttle Position (TP) Multiplier: Richens or leans the fuel based on throttle opening.

· Manifold Absolute Pressure (MAP) Multiplier: Richens or leans the fuel based on manifold pressure, either vacuum or boost.

· Transitional Fuel Enrichment (TFE): This richens the fuel when opening the throttle, like a carburetor’s accelerator pump.

· Injector Offset Adder: This corrects for injector offset; the time it takes for the injector to open.

· Programmable Output Fuel Multiplier: Richens or leans the fuel based on the output state of each of the four available output channels. What you hook up to these channels is up to you.

· Analog Input Fuel Multiplier: Richens or leans the fuel based on the input voltage of each of the three available channels. You decide what you want these channels connected to.

Setting up each of these fuel control parameters means the EPEC wants to know when the enrichment (or leaning) should begin, how much enrichment you want and when to turn it off. It might ask for an either/or decision, such as, do you want this form of enrichment if these other three enrichments are already underway? It can be complex, as there are many layers to go through. There is a stock default setting saved so you can get back to a standard calibration with minimal keystrokes. Also, you can save several custom calibrations of your own to save reinventing the calibration.

The spark functions are:

· Idle Spark Adder/Subtractor: Advances or retards spark based on rpm at idle.

· Part Throttle Spark Adder/Subtractor: Advances or retards spark based on rpm at part throttle.

· Wide Open Throttle (WOT) Spark Adder/Subtractor: Advances or retards spark based on rpm at WOT.

· Air Charge Temperature (ACT) Spark Adder/Subtractor: Advances or retards spark based on air inlet temperature.

· Engine Coolant Temperature (ECT) Spark Adder/Subtractor: Advances or retards spark based on engine temperature.

· Barometric Pressure (BP) Spark Adder/Subtractor: Advances or retards spark based on barometric pressure.

· Throttle Position (TP) Spark Adder/Subtractor: Advances or retards spark based on throttle opening

· Manifold Absolute Pressure (MAP) Spark Adder/Subtractor: Advances or retards spark based on manifold pressure (load). Requires fitment of a GM MAP sensor.

· Boost Retard Subtractor: Retards spark in degrees per lb. of boost based on rpm.

· Nitrous Spark Retard Subtractor: Retards spark during nitrous oxide operation (two stages).

· Programmable Output Spark Adder/Subtractor: Advances or retards spark based on the output state of each programmable output channel. You select what device these channels are connected to.

· Analog Input Spark Adder/Subtractor: Advances or retards spark based on the input voltage of each analog input channel. You decide what device to connect to these channels.

EPEC offers a 16-channel data logger that is fully programmable, so you can choose what data to log, at what time, for how long and through which limits. The amount of memory in the EPEC is limited, of course, so the more functions logged and the longer information is logged, the fewer data points and the slower the hertz (points per second) you'll get. There is plenty of memory for logging drag strip runs in useful time frames with many parameters, however. Essentially any data the EEC takes in can be logged, plus whatever you hook to the four programmable outputs and three analog inputs. By looking at things like the mass air meter and oxygen sensor signals, along with the usual rpm measurements, interesting phenomenon and correlations can be seen.

Because EPEC is already monitoring rpm and has its own clock, it's a cinch to let it control nitrous solenoids or other time/rpm dependent gear. Thus it makes a full-function nitrous controller. There's also a spare I/O wire from EPEC which can turn on something as simple as a bulb (Make your own shift light?), or any other device. Sam believes racers will use it to turn on a second data logger rigged to chassis inputs such as driveshaft speed, suspension travel and so on.

EPEC can also perform all ignition boost retard functions. This requires a MAP sensor (also useful for data logging). It recognizes the common GM MAP sensor, so the hardware is easily available. EPEC can also easily handle rev-limiting duties under its regular menu, including two-step functions.

Source:

Ford Motorsport SVO

44050 N. Groesbeck Highway

Clinton Township, MI 48036-1108

(810) 468-1356

REFERENCE

Compiled by Tom Cloud <cloud@peaches.ph.utexas.edu>
(font is Courier New)

EEC APPLICATIONS

A9L is the most common 89-93 MAF 5-speed computer catch code.

T4M0 is the most common 94-95 MAF 5-speed/E0D computer catch code.

J4J1 is the catch code on 94-95 Cobra computers.

ZA0 is the catch code used on the Cobra-R.

AKC0 is used by FMS for the Lightning Mass Air conversion.

engine	vehicle	year	efi type	xmsn	diff	Code	Calibration	Part Number
5.0	truck CA	95	MAF	E4OD	4.1			F5TF-12A650-HB
5.?	truck CA		MAF	E4OD	3.55			F5TF-12A650-GB
5.8	truck CA		MAF	E4OD	4.1			F5TF-12A650-BYA
5.8	Bronco,F-x50					39D1		
2.9	Scorpio					7GYA		
1.9	Escort					8AM		
1.9	Escort					8BB		
2.3	Mustang					8CC		
2.3	Tempo					8DN		
2.9	Ranger					8DR		
2.9	Scorpio					8GHB		
5.0	T’Bird					8KC		
5.0	Mustang		MAF			8LD		
2.9	Ranger					8ML		
3.0	Taurus	88				8NC		E9AF-14A624-AA
5.0	Bronco					8PZ		
5.0	Bronco					8PZ		
460	F350					8SE		
5.0	Bronco	88	SD	5-spd	3.55	8TP		
2.3	T’Bird Turbo					8UA		
5.8	Bronco,F-x50					A0C3		
5.0	F-150	91	SD	E4OD	3.55	A1C		
4.0	Ranger/Explr					A1S		
4.6	Crown Vic					A2J1		
5.8	Bronco,F-x50					A2Z		
5.8	Bronco,F-x50					A2Z1		
5.0	Mustang	93		5-spd	2.73	A3M	3-21A-R10	F3ZF-12A650-DA

Table 63: PCM Applications

engine	vehicle	year	efi type	xmsn	diff	Code	Calibration	Part Number
5.0	Mustang		MAF	5-spd	2.73	A3M1		
5.0	Mustang	89-93	MAF	5-spd		A9L		E9ZF-12A650-AA
5.0	Mustang		MAF			A9M		
5.0	Mustang		MAF			A9P		
5.0	Mustang / CA		MAF	5-spd		A9S		
5.0	Mustang	89	MAF			A9T		
1.9	Escort					AA2		
1.9	Escort					AB2		
5.0	T’Bird					AB2		
1.9	Escort					AB3		
3.0	Ranger					ACE1		
4.0	Ranger/Explr					ADZ1		
1.9	Escort					AF1		
1.9	Escort					AH1		
5.8	Lightning FMS	n/a	MAF	E4OD		AKC0		F8TF-12A650-NA
4.0	Ranger/Explr					ANY1		
3.8	T’Bird SC					B9A1		
3.0	Taurus SHO					B9B		
3.0	Taurus SHO					B9B1		
3.8	Cougar					B9L1		
3.8	T’Bird					B9L2		
4.0	Ranger/Explr					BAT1		
5.8	Bronco,F-x50					BTQ		
3.8	T’Bird SC					C0S		
4.0	Ranger/Explr					C1J		
5.8	Bronco,F-x50					C1Z		
5.0	Bronco					C2M1		
5.8	Bronco,F-x50					C2M1		
4.6	Crown Vic					C2Z3		
4.6	Crown Vic					C3N3		
5.8	Lightning			E4OD		C3P1		
5.8	Lightning			E4OD		C3P2		
5.0	Mustang	93		auto		C3W		F3ZF-12A650-FA
5.0	Mustang	93	MAF	auto		C3W1		F3ZF-12A650-FB
2.8	Ranger					C9B		
2.9	Ranger					C9E1		
2.9	Ranger					C9M		
3.0	Cougar					CE		
4.0	Ranger/Explr					COW1		
5.8	Bronco,F-x50					D1X		
5.0	T’Bird					D2L		
5.0	Mustang				3.08	D3D		F3ZF-12A650-EA
5.0	Mustang				3.08	D3D1		F3ZF-12A650-EB
3.0	Taurus					D9C		
3.0	Taurus					D9C1		
5.8	Bronco,F-x50					D9D1		

Table 63: PCM Applications

engine	vehicle	year	efi type	xmsn	diff	Code	Calibration	Part Number
5.8	Bronco,F-x50					D9L1		
	MK7					D9S		
5.0	Mustang	87	SD/SFI			DA1		E7SF-12A650-A1B
460	Van					DAD		
5.0	Mustang					DC		
5.0	Mustang					DE		
5.0	T’Bird					DG1		
4.6	Crown Vic					DH		
5.0	Mustang					DX3		
5.8	Bronco,F-x50					E0D		
4.0	Ranger/Explr					E0E		
4.0	Ranger/Explr					E0L		
5.0	T’Bird					E1X		
4.6	Crown Vic					E3Y2		
1.9	Escort					F1X		
2.3	Mustang					FB2		
2.3	Mustang SVO					FB2		
5.8	Bronco,F-x50					FK1		
5.0	Mustang					GJ1		
5.8	Bronco,F-x50					GT		
5.0	T’Bird					H2M		
5.0	T’Bird					H2M1		
3.2	Taurus SHO					H3Z		
4.0	Ranger/Explr					HAG0		
2.9	Ranger	87	SD	5-spd		HD		
460	F350					J2C1		
3.0	Ranger					J2Z		
5.0	Cobra	94-95				J4J1		
5.8	Bronco	95	SD	E4OD	3.55	JOG1		
4.0	Ranger/Explr					K1P0		
5.0	T’Bird					KF		
	Probe V6					KLO7		
4.0	Ranger/Explr					L0D		
3.0	Taurus SHO					L0S		
5.0	Bronco					L12D		
1.9	Escort					L1X		
4.6	Crown Vic					L2W		
2.3	T’Bird Turbo					LA		
2.3	T’Bird Turbo					LA2		
2.3	T’Bird Turbo					LA3		
2.3	T’Bird Turbo					LB2		
2.3	T’Bird Turbo					LB3		
2.9	Ranger					LDP1		
3.8	T’Bird SC					LOE1		
	MK7					M1L1		
4.6	Crown Vic					M2C		

Table 63: PCM Applications

engine	vehicle	year	efi type	xmsn	diff	Code	Calibration	Part Number
3.0	Ranger					M2T		
3.8	T’Bird SC					M2Y		
1.9	Escort					M2Z		
5.0	T’Bird					MC2		
5.0	G.Marquis					MN		
3.0	Ranger					MOM2		
3.8	T’Bird					MP		
4.0	Ranger/Explr					NAP2		
4.0	Ranger/Explr					OLD2		
4.0	Ranger/Explr					P0X0		
5.0	T’Bird					P3M		
4.0	Ranger/Explr					PAN1		
2.3	Mustang SVO					PC1		
2.3	Mustang SVO					PE		
2.3	Merkur Turbo					PF2		
2.3	Merkur Turbo					PF3		
2.3	Mustang SVO					PJ		
2.3	Mustang SVO					PK		
2.3	Mustang SVO					PK1		
4.0	Ranger/Explr					RAT1		
2.9	Ranger					RM2		
2.9	Bronco II	86	SD	A4LD		RP		
3.8	LTD					SX		
2.0	Probe 16V					T		
5.0	Econoline					T2T		
5.0	Mustang	94-95	MAF	EOD		T4MO		
2.3	T’Bird Turbo					TA		
2.3	Mustang SVO					TE		
2.3	T’Bird Turbo					TE		
2.3	T’Bird Turbo					TF		
5.8	Bronco,F-x50					U2U1		
3.8	T’Bird SC					U2Y		
5.0	Mustang					U4PO		
	Explorer	86				UB		E6EF-12A650-B1A
1.9	Escort	86				UB		
4.0	Ranger/Explr					UMP1		
4.0	Ranger/Explr					VAN		
4.0	Ranger/Explr					VET1		
5.0	Mustang	86	SFI			VH2		E6SF-12A650-H1C
2.3	Mustang SVO					VJ1		
5.0	Mustang					VJ1		
5.0	Mustang					VM1		
5.0	Mustang					VR1		
	diesel pickup	89				W1D		F1TF-12B565-CA
1.9	Escort					W1E		
3.8	T’Bird SC					W1M		

Table 63: PCM Applications

engine	vehicle	year	efi type	xmsn	diff	Code	Calibration	Part Number
5.0	Bronco					W2J		
5.8	Bronco,F-x50					W2J		
460	F350					W2T		
3.0	Taurus SHO					W2Z		
	MK8					W3Z2		
3.8	T'Bird SC					W4D2		
4.0	Ranger/Explr					X0A		
5.8	Bronco,F-x50					X0P		
3.8	T'Bird					X1A2		
3.8	T'Bird SC					X1A2		
3.0	Taurus SHO					X2J		
	XR7					X2P		
4.0	Ranger/Explr					X2T2		
5.0	Cobra	93				X3Z		FEZF-12A650-CA
4.0	Ranger/Explr					YAM1		
3.8	T'Bird SC					Z1Z2		
4.0	Ranger/Explr					Z2C2		
5.8	Bronco,F-x50					Z2D1		
3.8	T'Bird					Z2U2		
	MK8					Z4H0		
5.8	Cobra-R					ZA0		
2.3	T'Bird Turbo					ZAA		
2.3	Mustang SVO					ZBA		
2.3	T'Bird Turbo					ZGA		

Table 63: PCM Applications

TERMS

Abbrev.	Meaning
A/C	Air Conditioning
ACCS	A/C Cycling Switch
ACC	A/C Clutch Compressor
ACT	Air Charge Temperature sensor
ACV	Thermactor Air Control Valve
AOD	Automatic Overdrive Transmission
AXOD	Automatic Transaxle Overdrive
BOO	Brake On/Off switch
BP	Barometric Pressure sensor
CANP	Canister Purge solenoid
CCO	Converter Clutch Override
CDR	Crankcase Depression Regulator
CFI	Central Fuel Injection
CID	Cylinder Identification sensor
CKT	Circuit
DIS	Direct Ignition System (see also EDIS, TFI)

Table 64: EEC Related Terms

Abbrev.	Meaning
DVOM	Digital Volt/Ohm Meter
E4OD	Electronic 4-speed OverDrive Transmission
ECA	Electronic Control Assembly (processor, computer) (see MCU)
ECM	Electronic Control Module (see MCU)
ECT	Engine Coolant Temperature
ECU	Electronic Control Unit (see MCU)
EDF	Electric Drive Fan relay assembly
EDIS	Electronic DIStributor (see also DIS, TFI)
EEC	Electronic Engine Control
EGO	Exhaust Gas Oxygen sensor (see HEGO)
EGR	Exhaust Gas Recirculation system
EGRC	EGR Control solenoid or system
EGRV	EGR Vent solenoid or system
EVP	EGR Position sensor
EVR	EGR Valve Regulator
FI	Fuel Injector or Fuel Injection
FDM	Fuel Delivery Module
FP	Fuel Pump
FPM	Fuel Pump Monitor
GND or GRND	Ground (see also PWR GND)
HEDF	High Speed Electro Drive Fan relay or circuit
HEGO	Heated EGO sensor
HEGOG	HEGO Ground circuit
HO	High Output
HSC	High Swirl Combustion, engine type
IDM	Ignition Diagnostic Module
IGN	Ignition system or circuit
INJ	Injector or Injection
ISC	Idle Speed Control
ITS	Idle Tracking Switch
KAM	Keep Alive Memory
KAPWR	Keep Alive Power
KOEO	Key On Engine Off
KOER	Key On Engine Running
KS	Knock Sensor
L	Liter(s)
LOS	Limited Operation Strategy (computer function)
LUS	Lock-Up Solenoid
MAF	Mass Air Flow sensor, meter or circuit
MA PFI	Mass Air Sequential Port Fuel Injection system
MCU	Microprocessor Control Unit
MIL	Malfunction Indicator Light
MPFI	Multi Port Fuel Injection
NDS	Neutral Drive Switch
NGS	Neutral Gear Switch
NPS	Neutral Pressure Switch

Table 64: EEC Related Terms

Abbrev.	Meaning
OCC	Output Circuit Check
OHC	Over Head Camshaft (engine type)
OSC	Output State Check
PCM	Powertrain Control Module
PFE	Pressure Feedback EGR sensor or circuit
PFI	Port Fuel Injection
PIP	Profile Ignition Pickup
PSPS	Power Steering Pressure Switch
PWR GND	Power Ground circuit (see also GND)
RWD	Rear Wheel Drive
SC	Super Charged (engine type)
SIG RTN	Signal Return circuit
SIL	Shift Indicator Light
SPOUT	Spark Output Signal from ECA
SS	Shift Solenoid circuit
STAR	Self Test Automatic Readout (test equipment)
STI	Self Test Input circuit
STO	Self Test Output circuit
TAB/TAD	Thermactor Air Bypass/Diverter Tandem solenoid valves
TAPS	Throttle Angle Position Sensor (see TP/TPS)
TFI	Thick Film Ignition system (see DIS, EDIS)
TGS	Top Gear Switch (cancels SIL operation in top gear)
THS	Transmission Hydraulic Switch
TP/TPS	Throttle Position Sensor
TTS	Transmission Temperature Switch
VAF	Vane Air Flow sensor or circuit
VAT	Vane Air Temperature
VBATT	Vehicle Battery Voltage
VM	Vane Meter
VOM	Analog Volt/Ohm Meter
VPWR	Vehicle Power supply voltage (regulated 10-14 volts)
VREF	Voltage Reference (ECA supplied reference voltage 4-6 volts)
VSC	Vehicle Speed Control sensor or signal
VSS	Vehicle Speed Sensor or signal
WAC	WOT A/C Cut-off switch or circuit
WOT	Wide Open Throttle

Table 64: EEC Related Terms

EEC-IV REFERENCE SOURCES:

The Engine/Emissions Diagnosis manual, available for specific models and years, covers all emissions related maintenance procedures. It is available from Helm, Inc., (800) 782-4356.

"How to Understand, Service, and Modify Ford Fuel Injection and Electronic Engine Control", by Charles O. Probst, published by Robert Bentley of Cambridge, MA, USA, ISBN 0-8376-0301-3. It is available from a Ford Motorsports dealers and Classic Motorbooks at (800) 826-6600. It gives an overview of the sensors, actuators, and control algorithms used by the EEC-IV, step-by-step diagnostic procedures, wiring diagrams and tips on hot-rodding EEC-IV cars.

SAE paper #820900, "EEC-IV Tomorrow’s Electronic Engine Controls Today", David Hagen & Dennis Wilkie, Ford Motor Co., Dearborn, MI

Intel 8096 (MCS-96) literature. One document, available from their web site, is "27006102.pdf" entitled "APPLICATION NOTE; AP-248; Using The 8096"; Order Number: 270061-002. There are many other related documents available from Intel -- including the use of the A-D converter, the implementation of "fuzzy logic", instrumenting and controlling automotive applications, and other topics.

AFTER-MARKET SUPPLIERS:

Connectors for the EEC are apparently proprietary also, though some have said they are available through Amp, Farnell and DigiKey.

There seem to be two channels of ECM availability:

1 - OEMs and the companies they authorize, who together provide remanufactured ECMs through dealer channels;

2 - and those involved in the remanufacturing of ECMs for the true automotive aftermarket.

- A1 Cardone
- Echlin
- Micro-Tech Automotive
- Standard Motor Parts

Some of these companies catalog and offer product (or repair service) on almost 800 different ECM configurations for Ford-made vehicles in the model years from 1977-1993. Some of these are consolidations of applications, where units have proven and tested to be comparable. Foreign made vehicles sold under the Ford nameplate would add to this population of ECMs, since the above count is only Ford units.

For an idea of what the EEC does, and what can be done with it, get a demo of Mike Wesley’s calibrator for the EEC-IV at:

http://www.tiac.net/users/goape/index.htm