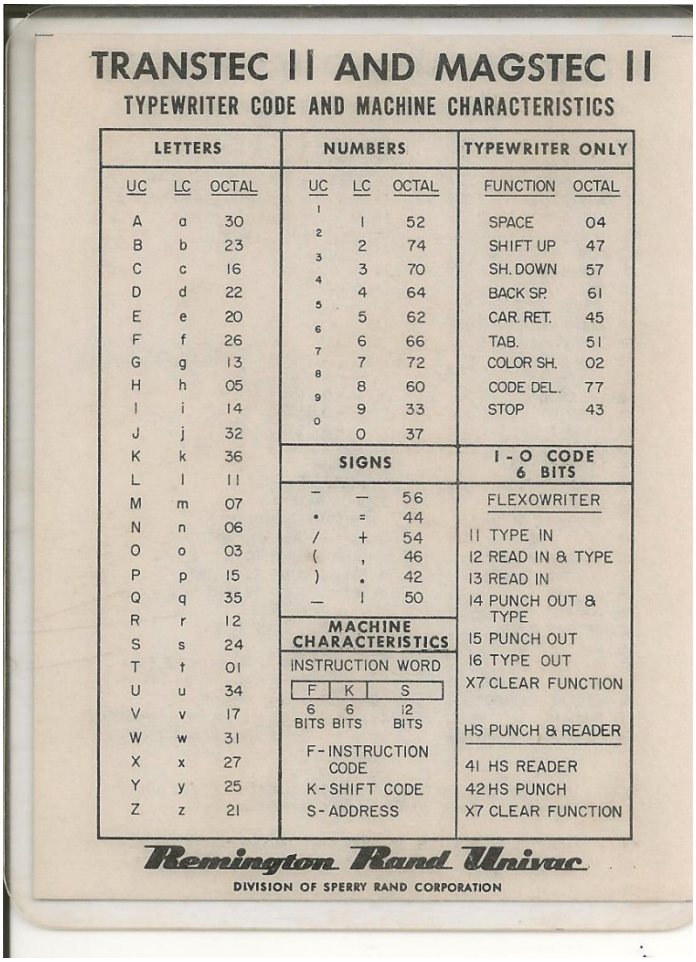
Repertoire Cards – 24 bits

INTRODUCTION

Programmers, field service engineers, test technicians, and design engineers all used pocket sized repertoire cards as a quick reference when trouble shooting hardware or when debugging software. As the VIP Club IT Legacy committee has been collecting documents, hardware artifacts, and career summaries; many repertoire cards have also been donated. This paper shows and discusses several repertoire card types for 24-bit computers and the sequence of these machines.

MAGSTEC - TRANSTEC

Take a look at the Transtec II and Magstec II card image at the right. In the middle, bottom is a small section labeled MACHINE CHARACTERISTICS. The 6 bit function code **F**, the 6 bit shift code **K**, and the 12 bit address code **S** make these two computers 24-bit Instruction Set Architecture (ISA) machines. The 12 bit addressing makes the immediate memory size 4,096 words. The other sections of this card lists the paper tape punch and print character codes.



TRANSTEC II AND MAGSTEC II

TYPEWRITER CODE AND MACHINE CHARACTERISTICS

LETTERS		
UC	LC	OCTAL
A	a	30
B	b	23
C	c	16
D	d	22
E	e	20
F	f	26
G	g	13
H	h	05
I	i	14
J	j	32
K	k	36
L	l	11
M	m	07
N	n	06
O	o	03
P	p	15
Q	q	35
R	r	12
S	s	24
T	t	01
U	u	34
V	v	17
W	w	31
X	x	27
Y	y	25
Z	z	21

NUMBERS		
UC	LC	OCTAL
1	1	52
2	2	74
3	3	70
4	4	64
5	5	62
6	6	66
7	7	72
8	8	60
9	9	33
0	0	37

SIGNS		
–	—	56
.	=	44
/	+	54
(	,	46
)	.	42
_	!	50

MACHINE CHARACTERISTICS

INSTRUCTION WORD

F	K	S
6 BITS	6 BITS	12 BITS

F - INSTRUCTION CODE
K - SHIFT CODE
S - ADDRESS

TYPEWRITER ONLY	
FUNCTION	OCTAL
SPACE	04
SHIFT UP	47
SH. DOWN	57
BACK SP.	61
CAR. RET.	45
TAB.	51
COLOR SH.	02
CODE DEL.	77
STOP	43

I - O CODE 6 BITS

FLEXOWRITER

11 TYPE IN
12 READ IN & TYPE
13 READ IN
14 PUNCH OUT & TYPE
15 PUNCH OUT
16 TYPE OUT
X7 CLEAR FUNCTION

HS PUNCH & READER

41 HS READER
42 HS PUNCH
X7 CLEAR FUNCTION

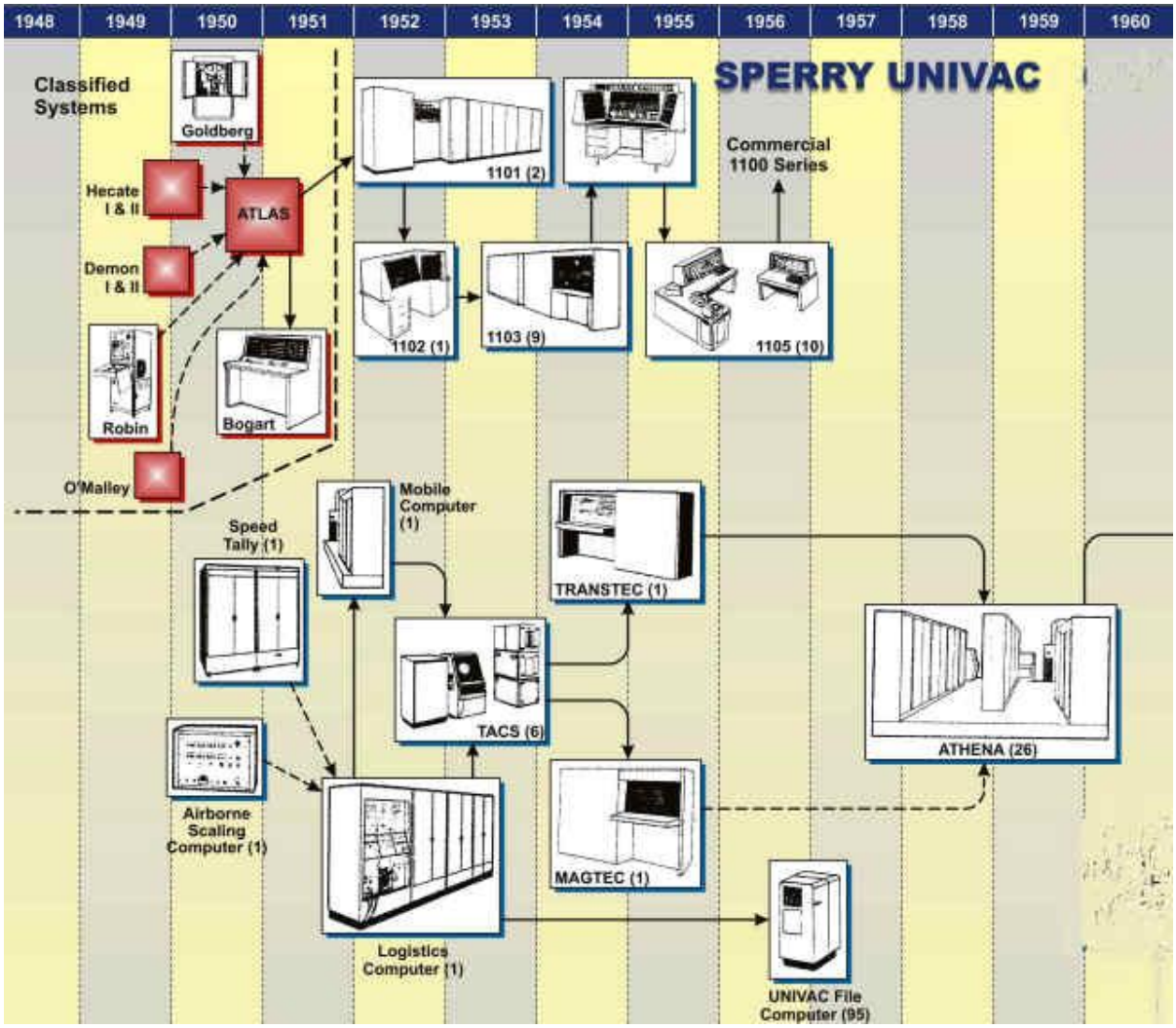
Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION

Harry Wise[1i] wrote:

1. "Transtec I and Magstec I were test beds for transistor and magnetic core logic. They were not computers but a rack of self testing logic.
2. Transtec II and Magstec II were 24 bit stored program computers. Each had 4,096 words of core memory. They were program compatible. They could be considered a follow on to the 24 bit Atlas/1101 vacuum tube computers. They remained around the plant for years being used for all sorts of things."

Warren Burrell[ii] wrote: "Several engineers [including myself] did try to graphically create values for the comparisons between magnetic logic and transistor logic after the tests and decisions had been made. The graphic showed a later ascendency of transistors though time. The two curves were very close for a period before the transistor gradually forged ahead."

Thus, transistor circuits became the normal design base instead of magnetic circuits. A genealogy chart excerpt below shows the chronological relationship of these two machines to other early ERA and UNIVAC computers. Their repertoire link back to the 24 bit ATLAS machine is not shown as that classified machine wasn't included UNIVAC documents until the mid 70's.

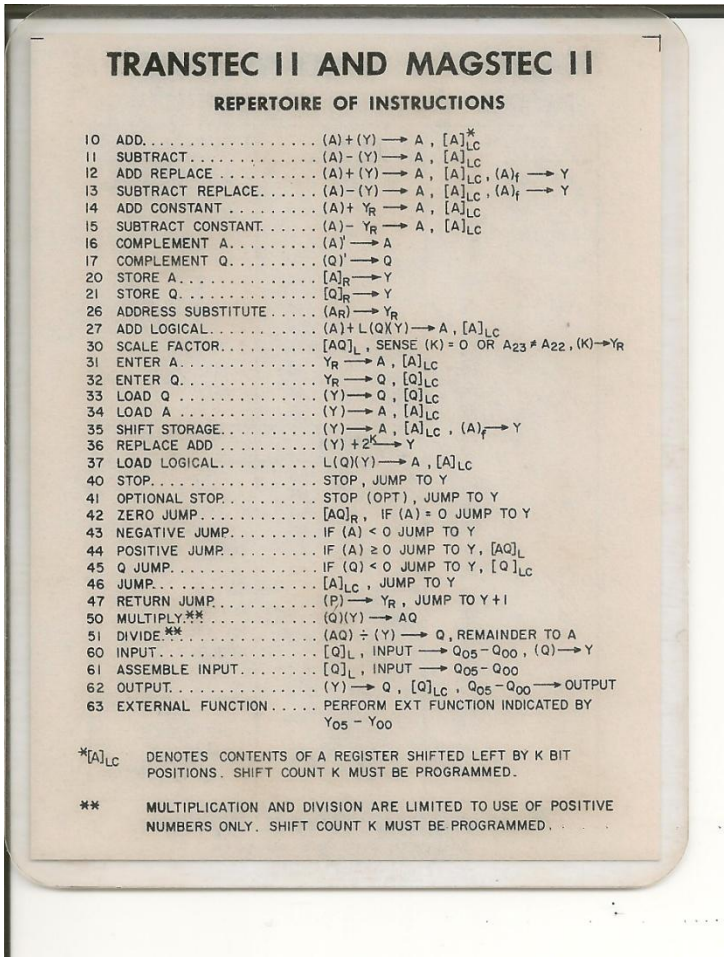


On this chart, the number in parenthesis after the computer name indicated the quantity built, however there was incomplete data available, for example there were five Goldberg units built and three 1102 computers [a 30-bit machine] delivered to the US Air Force at Arnold Research Center, see Warren Burrell's article on the People, A-B web site page. Above the 1103 [a 36-bit machine] is an unlabeled computer, the 1103A also called the Univac Scientific – 16 of those were built and delivered. The 1103 machines had a combination of rotating drum and core memories.

This author does not know the specific relationship of the Tactical Air Command Systems computers which are shown as preceding the Transtec/Magtec test machines. The Transtec instruction repertoire is listed on the other side of the repertoire card as shown on the next page.

Some of the symbols used on the repertoire card are:

- **A** is the 24 bit accumulator register,
- **Q** is the 24 bit quotient register,
- **(A)** represents the contents of the accumulator,
- **(Y)** is the 24 bit contents of memory address **S** as noted on the other side of the card,
- $\mathbf{Y_R}$ is a 12-bit operand of some instructions, the **S** bits of the instruction, and
- **P** is the 12 bit program address counter, incremented by 1 after fetching an address.



TRANSTEC 11 AND MAGSTEC 11

REPERTOIRE OF INSTRUCTIONS

10	ADD	$(A)+(Y) \longrightarrow A$, $[A]^{*}_{LC}$
11	SUBTRACT	$(A)-(Y) \longrightarrow A$, $[A]_{LC}$
12	ADD REPLACE	$(A)+(Y) \longrightarrow A$, $[A]_{LC}$, $(A)_f \longrightarrow Y$
13	SUBTRACT REPLACE	$(A)-(Y) \longrightarrow A$, $[A]_{LC}$, $(A)_f \longrightarrow Y$
14	ADD CONSTANT	$(A)+Y_R \longrightarrow A$, $[A]_{LC}$
15	SUBTRACT CONSTANT	$(A)-Y_R \longrightarrow A$, $[A]_{LC}$
16	COMPLEMENT A	$(A)' \longrightarrow A$
17	COMPLEMENT Q	$(Q)' \longrightarrow Q$
20	STORE A	$[A]_R \longrightarrow Y$
21	STORE Q	$[Q]_R \longrightarrow Y$
26	ADDRESS SUBSTITUTE	$(A_R) \longrightarrow Y_R$
27	ADD LOGICAL	$(A)+L(Q(Y) \longrightarrow A$, $[A]_{LC}$
30	SCALE FACTOR	$[AQ]_L$, SENSE $(K) = 0$ OR $A_{23} \neq A_{22}$, $(K) \longrightarrow Y_R$
31	ENTER A	$Y_R \longrightarrow A$, $[A]_{LC}$
32	ENTER Q	$Y_R \longrightarrow Q$, $[Q]_{LC}$
33	LOAD Q	$(Y) \longrightarrow Q$, $[Q]_{LC}$
34	LOAD A	$(Y) \longrightarrow A$, $[A]_{LC}$
35	SHIFT STORAGE	$(Y) \longrightarrow A$, $[A]_{LC}$, $(A)_f \longrightarrow Y$
36	REPLACE ADD	$(Y) + 2^{K} \longrightarrow Y$
37	LOAD LOGICAL	$L(Q)(Y) \longrightarrow A$, $[A]_{LC}$
40	STOP	STOP , JUMP TO Y
41	OPTIONAL STOP	STOP (OPT) , JUMP TO Y
42	ZERO JUMP	$[AQ]_R$, IF $(A) = 0$ JUMP TO Y
43	NEGATIVE JUMP	IF $(A) < 0$ JUMP TO Y
44	POSITIVE JUMP	IF $(A) \geq 0$ JUMP TO Y , $[AQ]_L$
45	Q JUMP	IF $(Q) < 0$ JUMP TO Y , $[Q]_{LC}$
46	JUMP	$[A]_{LC}$, JUMP TO Y
47	RETURN JUMP	$(P) \longrightarrow Y_R$, JUMP TO Y + 1
50	MULTIPLY**	$(Q)(Y) \longrightarrow AQ$
51	DIVIDE**	$(AQ) \div (Y) \longrightarrow Q$, REMAINDER TO A
60	INPUT	$[Q]_L$, INPUT $\longrightarrow Q_{05} - Q_{00}$, $(Q) \longrightarrow Y$
61	ASSEMBLE INPUT	$[Q]_L$, INPUT $\longrightarrow Q_{05} - Q_{00}$
62	OUTPUT	$(Y) \longrightarrow Q$, $[Q]_{LC}$, $Q_{05} - Q_{00} \longrightarrow$ OUTPUT
63	EXTERNAL FUNCTION	PERFORM EXT FUNCTION INDICATED BY $Y_{05} - Y_{00}$

*$[A]_{LC}$ DENOTES CONTENTS OF A REGISTER SHIFTED LEFT BY K BIT POSITIONS. SHIFT COUNT K MUST BE PROGRAMMED.

** MULTIPLICATION AND DIVISION ARE LIMITED TO USE OF POSITIVE NUMBERS ONLY. SHIFT COUNT K MUST BE PROGRAMMED.

Thus, the first instruction, 10 ADD, adds the contents of memory address **Y** to the contents of the accumulator **A**, leaving the left shifted results in the accumulator. Three variations of the ADD are the ADD REPLACE, the ADD CONSTANT, and the ADD LOGICAL instruction. The COMPLEMENT instructions would 'in effect' change a positive number to negative or vice versa. The ADD LOGICAL is unique in that it used the bit pattern 'preloaded' in the **Q** register as a mask to combine specific bit fields of **A** and **Y**. A common slang term for this type of operation was 'bit fiddling.' The **A** and **Q** registers become a 48-bit register for the MULTIPLY and DIVIDE instructions.

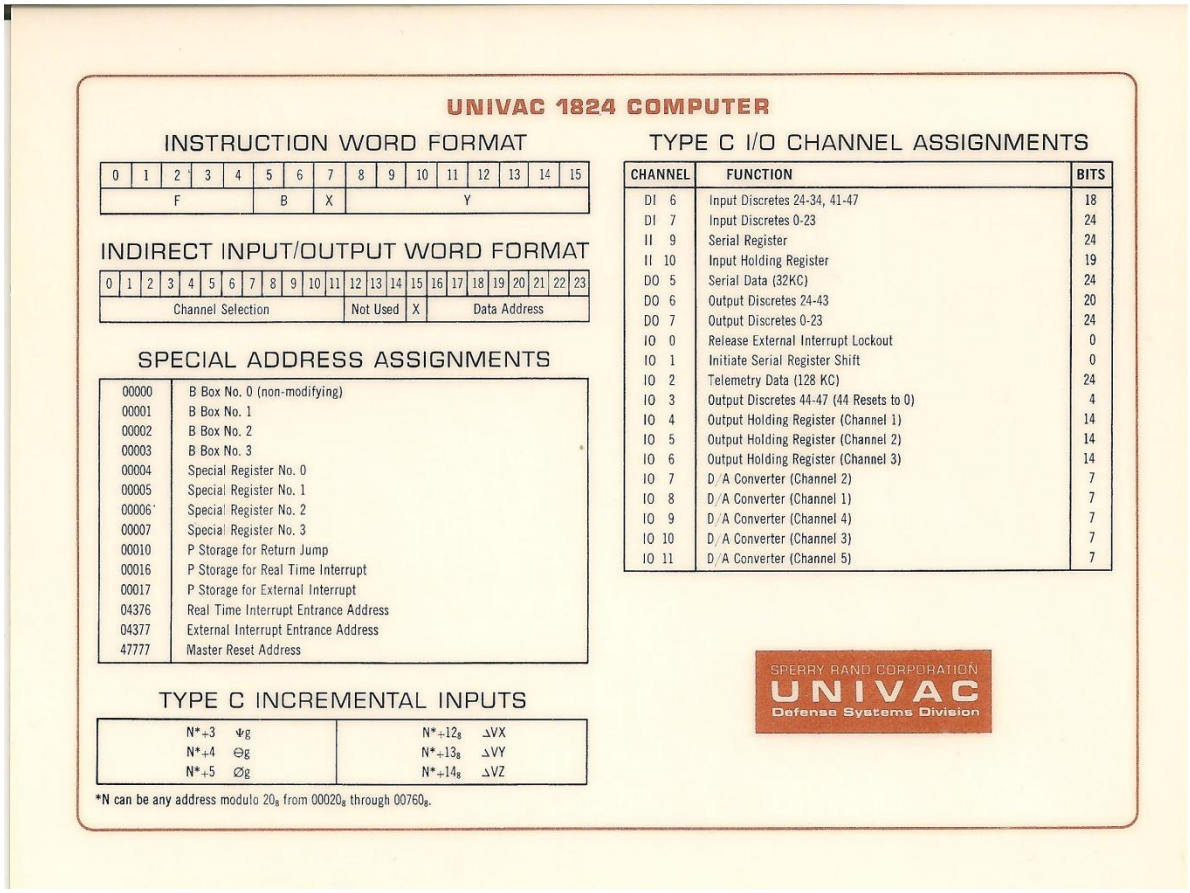
Branch Instructions: Instructions 42 and 45 test the content of a register, if no bits are set, load **P** with $\mathbf{Y_R}$. If any bit is set, the next instruction comes from the already incremented **P** register. Instructions 43 and 44 are used to test the positive or negative result of the previous operation. This is where mathematicians get involved! In a fixed length machine, the left most bit of the accumulator is used as the sign bit, i.e. a 0 means positive number and a 1 means a negative number. With this said, there are 23 bits of accuracy in this machine. If two positive numbers are added and the 'carry' results in a 1 in the 24^{th} bit – the result is an 'overflow' which must be dealt with within the logic of the software program. Op code 30, SCALE FACTOR, is also used to determine unique register conditions as it tests whether the **A** register bits 22 and 23 are not equal to each other.

Instruction code 47, RETURN JUMP, provides the main stream program with the capability of using sub-routines and returning back to the place calling the subroutine. Instruction code 63, EXTERNAL FUNCTION, is used to control the Flexo-writer input and output modes as well as the high speed punch.

UNIVAC 1824 COMPUTER

The 1824 was a space-borne version of the Athena rocket/missile guidance computer. As we look at the card image below we first see a 16 bit instruction word format. Five bits for the function code (f), two bits for the index register (B tube) selection, one bit as an extension (x) selector, and eight bits (Y) as the operand or operand address selector.

The second item on the card is the Indirect input/output word format, 12 bits for the 'Channel Selection, three unused bits then an x bit and 8 bits for data address. The x bit and 8-bits for Data Address appear to line up with the operand fields of the instruction word format. At the top right of the card are the Type C I/O Channel Assignments. The channel column has up to four 8ctal characters, the same number of bits as the Indirect input/output word format. These channels provide the computer (programmer) access to discrete, digital to Analog conversions, and holding registers.



UNIVAC 1824 COMPUTER

INSTRUCTION WORD FORMAT

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F					B		X	Y							

INDIRECT INPUT/OUTPUT WORD FORMAT

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Channel Selection												Not Used			X	Data Address							

SPECIAL ADDRESS ASSIGNMENTS

00000	B Box No. 0 (non-modifying)
00001	B Box No. 1
00002	B Box No. 2
00003	B Box No. 3
00004	Special Register No. 0
00005	Special Register No. 1
00006	Special Register No. 2
00007	Special Register No. 3
00010	P Storage for Return Jump
00016	P Storage for Real Time Interrupt
00017	P Storage for External Interrupt
04376	Real Time Interrupt Entrance Address
04377	External Interrupt Entrance Address
47777	Master Reset Address

TYPE C INCREMENTAL INPUTS

N*+3	Ψg	N*+12$_8$	ΔVX
N*+4	Θg	N*+13$_8$	ΔVY
N*+5	Øg	N*+14$_8$	ΔVZ

*N can be any address modulo 20$_8$ from 00020$_8$ through 00760$_8$.

TYPE C I/O CHANNEL ASSIGNMENTS

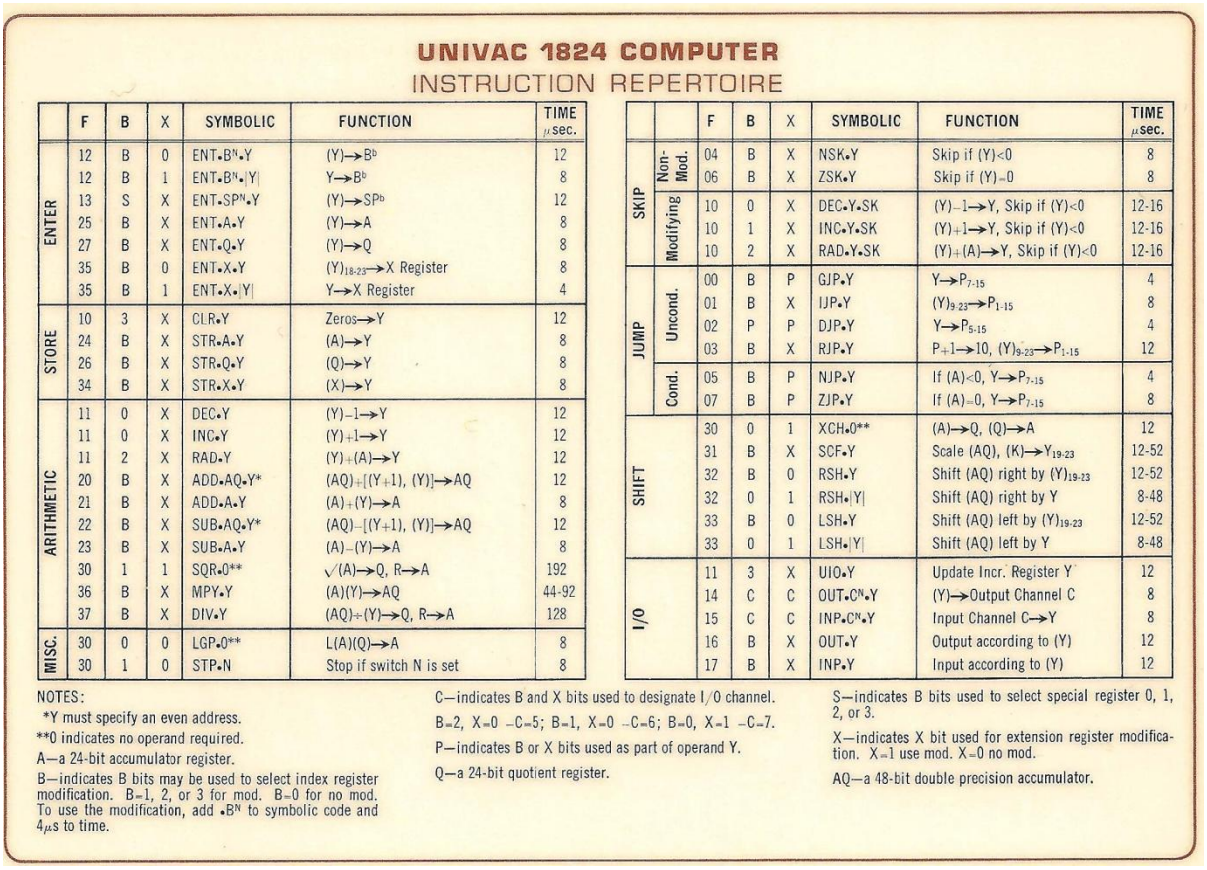
CHANNEL	FUNCTION	BITS
DI 6	Input Discretes 24-34, 41-47	18
DI 7	Input Discretes 0-23	24
II 9	Serial Register	24
II 10	Input Holding Register	19
DO 5	Serial Data (32KC)	24
DO 6	Output Discretes 24-43	20
DO 7	Output Discretes 0-23	24
IO 0	Release External Interrupt Lockout	0
IO 1	Initiate Serial Register Shift	0
IO 2	Telemetry Data (128 KC)	24
IO 3	Output Discretes 44-47 (44 Resets to 0)	4
IO 4	Output Holding Register (Channel 1)	14
IO 5	Output Holding Register (Channel 2)	14
IO 6	Output Holding Register (Channel 3)	14
IO 7	D/A Converter (Channel 2)	7
IO 8	D/A Converter (Channel 1)	7
IO 9	D/A Converter (Channel 4)	7
IO 10	D/A Converter (Channel 3)	7
IO 11	D/A Converter (Channel 5)	7

SPERRY RAND CORPORATION
UNIVAC
Defense Systems Division

The Special Address Assignments are listed next on the left of this side of the card. The left column of this table show five octal characters, 15 bits which would imply 32k of memory. The address assignments listed imply that these are in a Random Access Memory, else the program couldn't function.

The obverse side of the card lists the instructions. There are a couple things of significance shown on this card compared to the older Transtec card. The Symbolic code gave the programmer to write ENT as a mnemonic for the action to enter a register with a value or the contents of an address. I personally remember in early 1964 when Ken Van Duren and John Heideman were developing the AS-1 assembler to interpret the symbolic codes from punched cards then generate executable octal code which would be punched onto paper tape for loading into machines.

Another column shows the execution time of of each instruction. Note that these are all in increments of 4 microseconds. The simplest instruction is the 34 code, STR Y->X which takes the 8-bit operand from the right of the instruction and places it into the extension register. 8 microsecond instructions take two memory cycles, one to fetch the instruction and one to fetch and act upon the operand. The small print below states that the instructions which have a **B** in the B column take an extra 4 microseconds because the contents of the designated B register must be fetched from memory.



UNIVAC 1824 COMPUTER

INSTRUCTION REPERTOIRE

	F	B	X	SYMBOLIC	FUNCTION	TIME μsec.
ENTER	12	B	0	ENT•B^N•Y	(Y)→B^b	12
	12	B	1	ENT•B^N•|Y|	Y→B^b	8
	13	S	X	ENT•SP^N•Y	(Y)→SP^b	12
	25	B	X	ENT•A•Y	(Y)→A	8
	27	B	X	ENT•Q•Y	(Y)→Q	8
	35	B	0	ENT•X•Y	$(Y)_{18-23}$→X Register	8
	35	B	1	ENT•X•|Y|	Y→X Register	4
STORE	10	3	X	CLR•Y	Zeros→Y	12
	24	B	X	STR•A•Y	(A)→Y	8
	26	B	X	STR•Q•Y	(Q)→Y	8
	34	B	X	STR•X•Y	(X)→Y	8
ARITHMETIC	11	0	X	DEC•Y	(Y)−1→Y	12
	11	0	X	INC•Y	(Y)+1→Y	12
	11	2	X	RAD•Y	(Y)+(A)→Y	12
	20	B	X	ADD•AQ•Y*	(AQ)+[(Y+1), (Y)]→AQ	12
	21	B	X	ADD•A•Y	(A)+(Y)→A	8
	22	B	X	SUB•AQ•Y*	(AQ)−[(Y+1), (Y)]→AQ	12
	23	B	X	SUB•A•Y	(A)−(Y)→A	8
	30	1	1	SQR•0**	$\sqrt{(A)}$→Q, R→A	192
	36	B	X	MPY•Y	(A)(Y)→AQ	44-92
	37	B	X	DIV•Y	(AQ)÷(Y)→Q, R→A	128
MISC.	30	0	0	LGP•0**	L(A)(Q)→A	8
	30	1	0	STP•N	Stop if switch N is set	8

		F	B	X	SYMBOLIC	FUNCTION	TIME μsec.
SKIP	Non-Mod.	04	B	X	NSK•Y	Skip if (Y)<0	8
		06	B	X	ZSK•Y	Skip if (Y)=0	8
	Modifying	10	0	X	DEC•Y•SK	(Y)−1→Y, Skip if (Y)<0	12-16
		10	1	X	INC•Y•SK	(Y)+1→Y, Skip if (Y)<0	12-16
		10	2	X	RAD•Y•SK	(Y)+(A)→Y, Skip if (Y)<0	12-16
JUMP	Uncond.	00	B	P	GJP•Y	Y→P_{7-15}	4
		01	B	X	IJP•Y	$(Y)_{9-23}$→P_{1-15}	8
		02	P	P	DJP•Y	Y→P_{5-15}	4
		03	B	X	RJP•Y	P+1→10, $(Y)_{9-23}$→P_{1-15}	12
	Cond.	05	B	P	NJP•Y	If (A)<0, Y→P_{7-15}	4
		07	B	P	ZJP•Y	If (A)=0, Y→P_{7-15}	8
SHIFT		30	0	1	XCH•0**	(A)→Q, (Q)→A	12
		31	B	X	SCF•Y	Scale (AQ), (K)→Y_{19-23}	12-52
		32	B	0	RSH•Y	Shift (AQ) right by $(Y)_{19-23}$	12-52
		32	0	1	RSH•|Y|	Shift (AQ) right by Y	8-48
		33	B	0	LSH•Y	Shift (AQ) left by $(Y)_{19-23}$	12-52
		33	0	1	LSH•|Y|	Shift (AQ) left by Y	8-48
I/O		11	3	X	UIO•Y	Update Incr. Register Y	12
		14	C	C	OUT•C^N•Y	(Y)→Output Channel C	8
		15	C	C	INP•C^N•Y	Input Channel C→Y	8
		16	B	X	OUT•Y	Output according to (Y)	12
		17	B	X	INP•Y	Input according to (Y)	12

NOTES:

*Y must specify an even address.

**0 indicates no operand required.

A—a 24-bit accumulator register.

B—indicates B bits may be used to select index register modification. B=1, 2, or 3 for mod. B=0 for no mod. To use the modification, add •B^N to symbolic code and 4μs to time.

C—indicates B and X bits used to designate I/O channel. B=2, X=0 –C=5; B=1, X=0 –C=6; B=0, X=1 –C=7.

P—indicates B or X bits used as part of operand Y.

Q—a 24-bit quotient register.

S—indicates B bits used to select special register 0, 1, 2, or 3.

X—indicates X bit used for extension register modification. X=1 use mod. X=0 no mod.

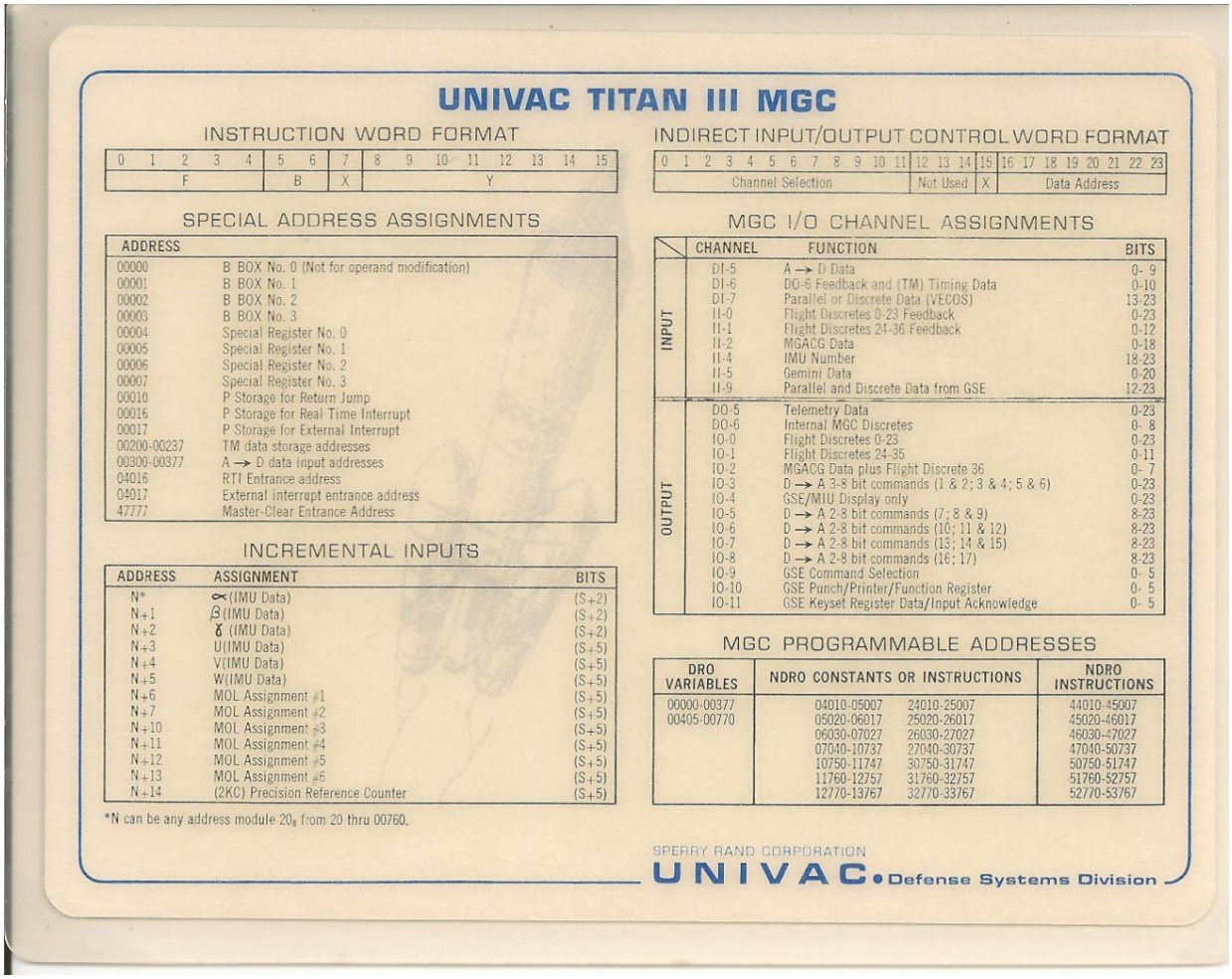
AQ—a 48-bit double precision accumulator.

The other thing that we see on this card is that the **A**ccumulator is 24 bits as is the **Q**uotient register. The Divide and Multiply Instructions as well as a couple of the shift instructions use these two as a 48 bit register.

The use of only 8 bits for Y (address) of most instructions only provides a 256 word range which means that the extension register provides the more significant bits of addresses, pointing toward selected blocks of memory.

UNIVAC TITAN III Missile Guidance COMPUTER (MGC)

A variation of the 1824 computer is the Titan computer. The instruction word format side of the repertoire card is quite similar however the input/output channel assignments are different and the programmable addresses table indicates locations of Non-Destructive Read Out addresses. (NDRO)



UNIVAC TITAN III MGC

INSTRUCTION WORD FORMAT

0 1 2 3 4	5 6	7	8 9 10 11 12 13 14 15
F	B	X	Y

INDIRECT INPUT/OUTPUT CONTROL WORD FORMAT

0 1 2 3 4 5 6 7 8 9 10 11	12 13 14	15	16 17 18 19 20 21 22 23
Channel Selection	Not Used	X	Data Address

SPECIAL ADDRESS ASSIGNMENTS

ADDRESS	
00000	B BOX No. 0 (Not for operand modification)
00001	B BOX No. 1
00002	B BOX No. 2
00003	B BOX No. 3
00004	Special Register No. 0
00005	Special Register No. 1
00006	Special Register No. 2
00007	Special Register No. 3
00010	P Storage for Return Jump
00016	P Storage for Real Time Interrupt
00017	P Storage for External Interrupt
00200-00237	TM data storage addresses
00300-00377	A → D data input addresses
04016	RTI Entrance address
04017	External interrupt entrance address
47777	Master-Clear Entrance Address

INCREMENTAL INPUTS

ADDRESS	ASSIGNMENT	BITS
N*	∝(IMU Data)	(S+2)
N+1	β(IMU Data)	(S+2)
N+2	δ (IMU Data)	(S+2)
N+3	U(IMU Data)	(S+5)
N+4	V(IMU Data)	(S+5)
N+5	W(IMU Data)	(S+5)
N+6	MOL Assignment #1	(S+5)
N+7	MOL Assignment #2	(S+5)
N+10	MOL Assignment #3	(S+5)
N+11	MOL Assignment #4	(S+5)
N+12	MOL Assignment #5	(S+5)
N+13	MOL Assignment #6	(S+5)
N+14	(2KC) Precision Reference Counter	(S+5)

*N can be any address module 20_8 from 20 thru 00760.

MGC I/O CHANNEL ASSIGNMENTS

	CHANNEL	FUNCTION	BITS
INPUT	DI-5	A → D Data	0- 9
	DI-6	DO-6 Feedback and (TM) Timing Data	0-10
	DI-7	Parallel or Discrete Data (VECOS)	13-23
	II-0	Flight Discretes 0-23 Feedback	0-23
	II-1	Flight Discretes 24-36 Feedback	0-12
	II-2	MGACG Data	0-18
	II-4	IMU Number	18-23
	II-5	Gemini Data	0-20
	II-9	Parallel and Discrete Data from GSE	12-23
OUTPUT	DO-5	Telemetry Data	0-23
	DO-6	Internal MGC Discretes	0- 8
	IO-0	Flight Discretes 0-23	0-23
	IO-1	Flight Discretes 24-35	0-11
	IO-2	MGACG Data plus Flight Discrete 36	0- 7
	IO-3	D → A 3-8 bit commands (1 & 2; 3 & 4; 5 & 6)	0-23
	IO-4	GSE/MIU Display only	0-23
	IO-5	D → A 2-8 bit commands (7; 8 & 9)	8-23
	IO-6	D → A 2-8 bit commands (10; 11 & 12)	8-23
	IO-7	D → A 2-8 bit commands (13; 14 & 15)	8-23
	IO-8	D → A 2-8 bit commands (16; 17)	8-23
	IO-9	GSE Command Selection	0- 5
	IO-10	GSE Punch/Printer/Function Register	0- 5
	IO-11	GSE Keyset Register Data/Input Acknowledge	0- 5

MGC PROGRAMMABLE ADDRESSES

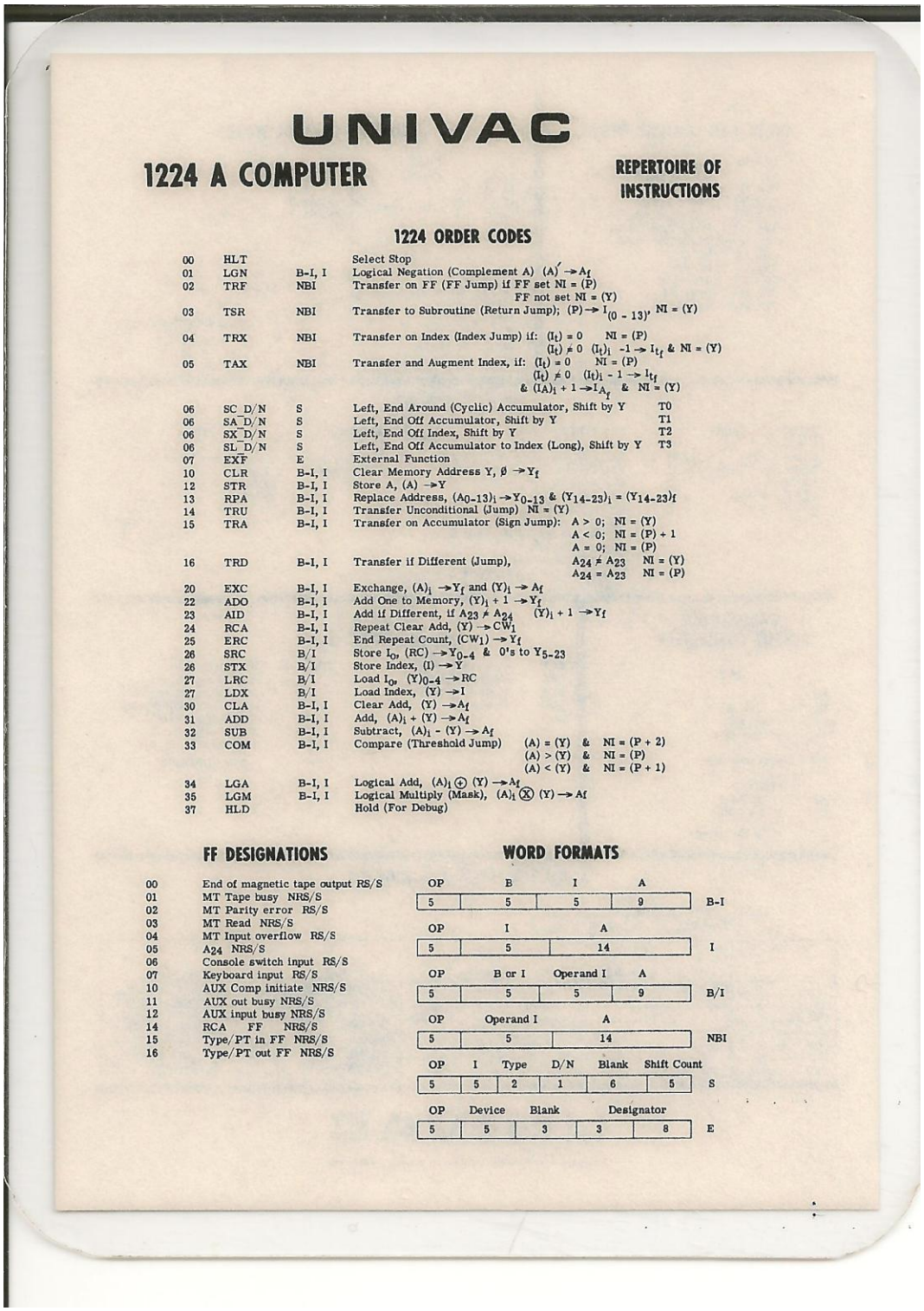
DRO VARIABLES	NDRO CONSTANTS OR INSTRUCTIONS		NDRO INSTRUCTIONS
00000-00377	04010-05007	24010-25007	44010-45007
00405-00770	05020-06017	25020-26017	45020-46017
	06030-07027	26030-27027	46030-47027
	07040-10737	27040-30737	47040-50737
	10750-11747	30750-31747	50750-51747
	11760-12757	31760-32757	51760-52757
	12770-13767	32770-33767	52770-53767

SPERRY RAND CORPORATION
UNIVAC • Defense Systems Division

The Instruction Repertoire side of the TITAN III card isn't shown here because it is virtually identical to the previously shown 1824 card.

UNIVAC Type 1224

The 1224 computer was developed for one of the 'Agencies' with a lot of bit fiddling instructions typically used in cryptology. There are 6 variations of the instruction word format, no divide!



UNIVAC

1224 A COMPUTER

REPERTOIRE OF INSTRUCTIONS

1224 ORDER CODES

00	HLT		Select Stop
01	LGN	B-I, I	Logical Negation (Complement A) $(A)' \rightarrow A_f$
02	TRF	NBI	Transfer on FF (FF Jump) if FF set NI = (P) FF not set NI = (Y)
03	TSR	NBI	Transfer to Subroutine (Return Jump); $(P) \rightarrow I_{(0-13)}$, NI = (Y)
04	TRX	NBI	Transfer on Index (Index Jump) if: $(I_t) = 0$ NI = (P) $(I_t) \neq 0$ $(I_t)_i - 1 \rightarrow I_{t_f}$ & NI = (Y)
05	TAX	NBI	Transfer and Augment Index, if: $(I_t) = 0$ NI = (P) $(I_t) \neq 0$ $(I_t)_i - 1 \rightarrow I_{t_f}$ & $(IA)_i + 1 \rightarrow IA_f$ & NI = (Y)
06	SC D/N	S	Left, End Around (Cyclic) Accumulator, Shift by Y T0
06	SA D/N	S	Left, End Off Accumulator, Shift by Y T1
06	SX D/N	S	Left, End Off Index, Shift by Y T2
06	SL D/N	S	Left, End Off Accumulator to Index (Long), Shift by Y T3
07	EXF	E	External Function
10	CLR	B-I, I	Clear Memory Address Y, $\emptyset \rightarrow Y_f$
12	STR	B-I, I	Store A, $(A) \rightarrow Y$
13	RPA	B-I, I	Replace Address, $(A_{0-13})_i \rightarrow Y_{0-13}$ & $(Y_{14-23})_i = (Y_{14-23})_f$
14	TRU	B-I, I	Transfer Unconditional (Jump) NI = (Y)
15	TRA	B-I, I	Transfer on Accumulator (Sign Jump): A > 0; NI = (Y) A < 0; NI = (P) + 1 A = 0; NI = (P)
16	TRD	B-I, I	Transfer if Different (Jump), $A_{24} \neq A_{23}$ NI = (Y) $A_{24} = A_{23}$ NI = (P)
20	EXC	B-I, I	Exchange, $(A)_i \rightarrow Y_f$ and $(Y)_i \rightarrow A_f$
22	ADO	B-I, I	Add One to Memory, $(Y)_i + 1 \rightarrow Y_f$
23	AID	B-I, I	Add if Different, if $A_{23} \neq A_{24}$ $(Y)_i + 1 \rightarrow Y_f$
24	RCA	B-I, I	Repeat Clear Add, $(Y) \rightarrow CW_1$
25	ERC	B-I, I	End Repeat Count, $(CW_1) \rightarrow Y_f$
26	SRC	B/I	Store I_o, $(RC) \rightarrow Y_{0-4}$ & 0's to Y_{5-23}
26	STX	B/I	Store Index, $(I) \rightarrow Y$
27	LRC	B/I	Load I_o, $(Y)_{0-4} \rightarrow RC$
27	LDX	B/I	Load Index, $(Y) \rightarrow I$
30	CLA	B-I, I	Clear Add, $(Y) \rightarrow A_f$
31	ADD	B-I, I	Add, $(A)_i + (Y) \rightarrow A_f$
32	SUB	B-I, I	Subtract, $(A)_i - (Y) \rightarrow A_f$
33	COM	B-I, I	Compare (Threshold Jump) (A) = (Y) & NI = (P + 2) (A) > (Y) & NI = (P) (A) < (Y) & NI = (P + 1)
34	LGA	B-I, I	Logical Add, $(A)_i \oplus (Y) \rightarrow A_f$
35	LGM	B-I, I	Logical Multiply (Mask), $(A)_i \otimes (Y) \rightarrow A_f$
37	HLD		Hold (For Debug)

FF DESIGNATIONS

00	End of magnetic tape output RS/S
01	MT Tape busy NRS/S
02	MT Parity error RS/S
03	MT Read NRS/S
04	MT Input overflow RS/S
05	A_{24} NRS/S
06	Console switch input RS/S
07	Keyboard input RS/S
10	AUX Comp initiate NRS/S
11	AUX out busy NRS/S
12	AUX input busy NRS/S
14	RCA FF NRS/S
15	Type/PT in FF NRS/S
16	Type/PT out FF NRS/S

WORD FORMATS

OP	B	I	A	
5	5	5	9	B-I

OP	I	A	
5	5	14	I

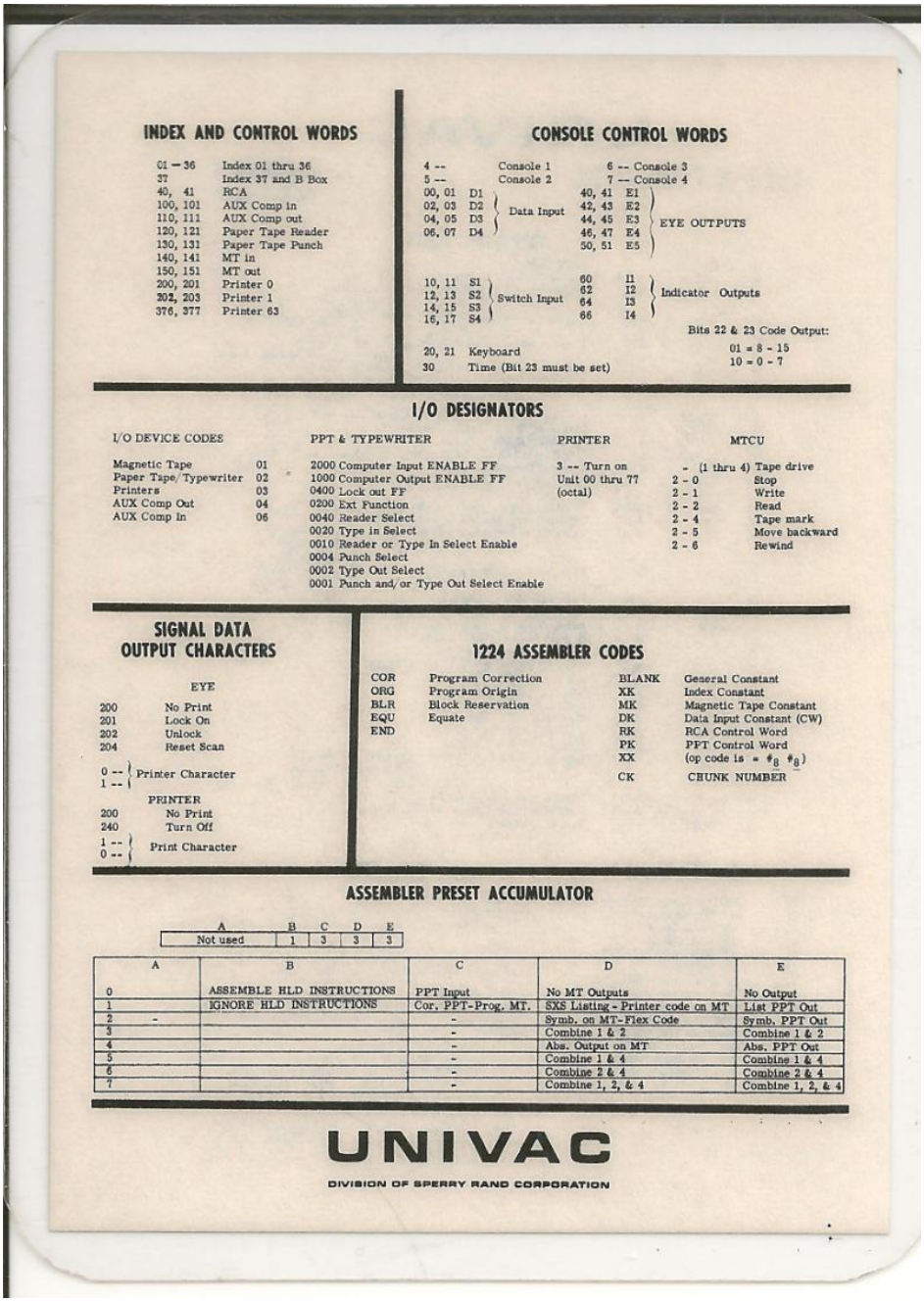
OP	B or I	Operand I	A	
5	5	5	9	B/I

OP	Operand I	A	
5	5	14	NBI

OP	I	Type	D/N	Blank	Shift Count	
5	5	2	1	6	5	S

OP	Device	Blank	Designator	
5	5	3	3	8	E

The obverse side of shows codes for the input/output devices as well as for the assembler.



INDEX AND CONTROL WORDS

01 – 36	Index 01 thru 36
37	Index 37 and B Box
40, 41	RCA
100, 101	AUX Comp in
110, 111	AUX Comp out
120, 121	Paper Tape Reader
130, 131	Paper Tape Punch
140, 141	MT in
150, 151	MT out
200, 201	Printer 0
202, 203	Printer 1
376, 377	Printer 63

CONSOLE CONTROL WORDS

4 -- Console 1
5 -- Console 2
6 -- Console 3
7 -- Console 4

Data Input: 00, 01 D1; 02, 03 D2; 04, 05 D3; 06, 07 D4

EYE OUTPUTS: 40, 41 E1; 42, 43 E2; 44, 45 E3; 46, 47 E4; 50, 51 E5

Switch Input: 10, 11 S1; 12, 13 S2; 14, 15 S3; 16, 17 S4

Indicator Outputs: 60 I1; 62 I2; 64 I3; 66 I4

Bits 22 & 23 Code Output:
01 = 8 - 15
10 = 0 - 7

20, 21 Keyboard
30 Time (Bit 23 must be set)

I/O DESIGNATORS

I/O DEVICE CODES		PPT & TYPEWRITER	PRINTER	MTCU	
Magnetic Tape	01	2000 Computer Input ENABLE FF	3 -- Turn on	- (1 thru 4)	Tape drive
Paper Tape/Typewriter	02	1000 Computer Output ENABLE FF	Unit 00 thru 77	2 - 0	Stop
Printers	03	0400 Lock out FF	(octal)	2 - 1	Write
AUX Comp Out	04	0200 Ext Function		2 - 2	Read
AUX Comp In	06	0040 Reader Select		2 - 4	Tape mark
		0020 Type in Select		2 - 5	Move backward
		0010 Reader or Type In Select Enable		2 - 6	Rewind
		0004 Punch Select			
		0002 Type Out Select			
		0001 Punch and/or Type Out Select Enable			

SIGNAL DATA OUTPUT CHARACTERS

EYE
200 No Print
201 Lock On
202 Unlock
204 Reset Scan

0 -- / 1 -- Printer Character

PRINTER
200 No Print
240 Turn Off

1 -- / 0 -- Print Character

1224 ASSEMBLER CODES

COR	Program Correction	BLANK	General Constant
ORG	Program Origin	XK	Index Constant
BLR	Block Reservation	MK	Magnetic Tape Constant
EQU	Equate	DK	Data Input Constant (CW)
END		RK	RCA Control Word
		PK	PPT Control Word
		XX	(op code is = ϕ_8 ϕ_8)
		CK	CHUNK NUMBER

ASSEMBLER PRESET ACCUMULATOR

A	B	C	D	E
Not used	1	3	3	3

A	B	C	D	E
0	ASSEMBLE HLD INSTRUCTIONS	PPT Input	No MT Outputs	No Output
1	IGNORE HLD INSTRUCTIONS	Cor. PPT-Prog. MT.	SXS Listing - Printer code on MT	List PPT Out
2	-	-	Symb. on MT-Flex Code	Symb. PPT Out
3		-	Combine 1 & 2	Combine 1 & 2
4		-	Abs. Output on MT	Abs. PPT Out
5		-	Combine 1 & 4	Combine 1 & 4
6		-	Combine 2 & 4	Combine 2 & 4
7		-	Combine 1, 2, & 4	Combine 1, 2, & 4

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

We'd welcome user inputs for more information about the 1224 machines and their applications. I do know that Don Mager led the 1224 machine logic design and that Dick Erdrich did most of the factory test programming. Ernie Lantto managed the 1224A hardware upgrades. Thanks, *LABenson*

[i] See http://vipclubmn.org/Computers.html#Comments, page 50 of our anthology.
[2] See http://vipclubmn.org/Articles/PreATHENA.pdf,