Signetics

N3002
Central Processing Element

Product Specification

Logic Products

FEATURES
- **45ns cycle time (typ)**
- **Easy expansion to multiple of 2 bits**
- **11 general purpose registers**
- **Full function accumulator**
- **Useful functions include:**
- **2's complement arithmetic**
  - **Logical AND, OR, NOT, exclusive-NOR**
  - **Increment, decrement**
  - **Shift left/shift right**
  - **Bit testing and zero detection**
  - **Carry look-ahead generation**
  - **Masking via K-bus conditioned locking allowing non-destructive testing of data in accumulator and scratchpad**
- **3 input buses**
- **2 output buses**
- **Control bus**

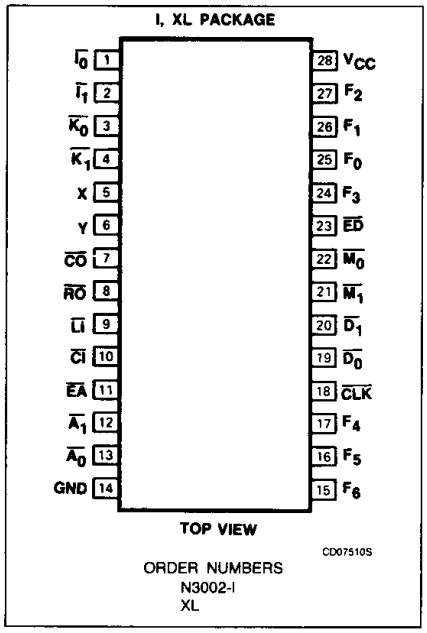
DESCRIPTION
The N3002 Central Processing Element (CPE) is one part of a bipolar microcomputer set. The N3002 is organized as a 2-bit slice and performs the logical and arithmetic functions required by microinstructions. A system with any number of bits in a data word can be implemented by using multiple N3002s, the N3001 microcomputer control unit, the N74S182 carry look-ahead unit and ROM or PROM memory.

FUNCTION TRUTH TABLE

FUNCTION GROUP	F_6	F_5	F_4
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

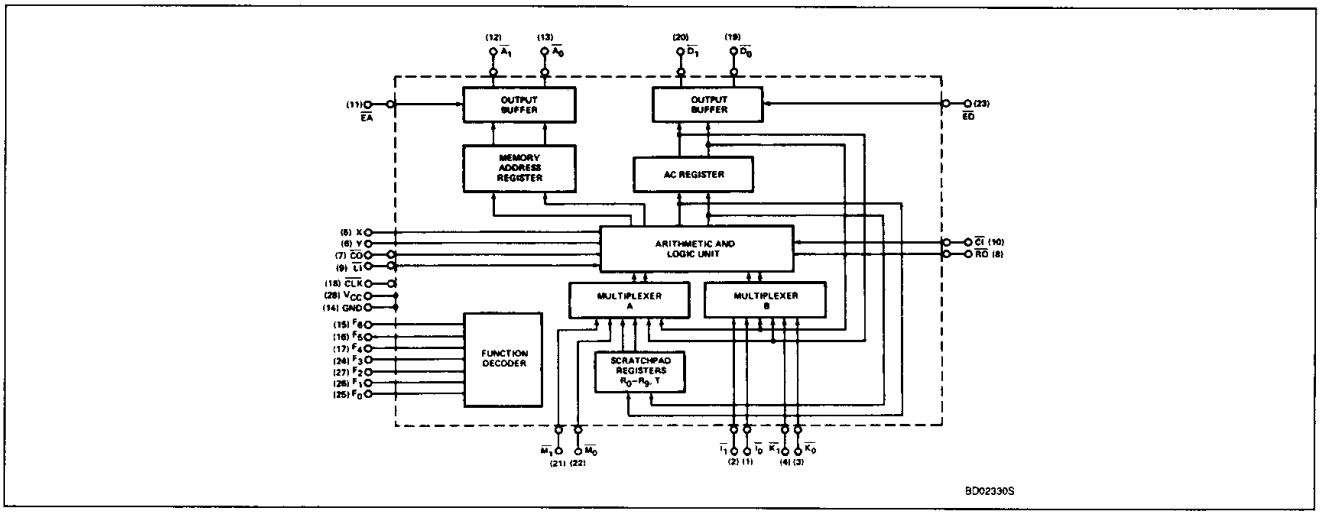
REGISTER GROUP	REGISTER	F_3	F_2	F_1	F_0
I	R_0	0	0	0	0
	R_1	0	0	0	1
	R_2	0	0	1	0
	R_3	0	0	1	1
	R_4	0	1	0	0
	R_5	0	1	0	1
	R_6	0	1	1	0
	R_7	0	1	1	1
	R_8	1	0	0	0
	R_9	1	0	0	1
	T	1	1	0	0
	AC	1	1	0	1
II	T	1	0	1	0
	AC	1	0	1	1
III	T	1	1	1	0
	AC	1	1	1	1

PIN CONFIGURATION



ORDER NUMBERS
N3002-I
XL

BLOCK DIAGRAM



Central Processing Element N3002

PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1, 2	$I_0 - I_1$	**External Bus Input:** The external bus inputs provide a separate input port for external input devices.	Active low
3, 4	$K_0 - K_1$	**Mask Bus Inputs:** The mask bus inputs provide a separate input port from the microprogram memory, to allow mask or constant entry	Active low
5, 6	X, Y	**Standard Carry Look-Ahead Cascade Outputs:** The cascade outputs allow high speed arithmetic operations to be performed when they are used in conjunction with the 74S182 Look-Ahead Carry Generator	Active high
7	CO	**Ripple Carry Out:** The ripple carry output is only disabled during shift right operations.	Active low Three-state
8	RO	**Shift Right Output:** The shift right output is only enabled during shift right operations.	Active low Three-state
9	LI	**Shift Right Input**	Active low
10	CI	**Carry Input**	Active low
11	EA	**Memory Address Enable Input:** When in the low state, the memory address enable input enables the memory address outputs ($A_0 - A_1$).	Active low
12 — 13	$A_0 - A_1$	**Memory Address Bus Outputs:** The memory address bus outputs are the buffered outputs of the memory address register (MAR).	Active low Three-state
14	GND	**Ground**	
14 — 17 24 — 27	$F_0 - F_6$	**Micro-Function Bus Inputs:** The micro-function bus inputs control ALU function and register selection.	Active high
18	CLK	**Clock Input**	
19 — 20	$D_0 - D_1$	**Memory Data Bus Outputs:** The memory data bus outputs are the buffered outputs of the full function accumulator register (AC).	Active low Three-state
21 — 22	$M_0 - M_1$	**Memory Data Bus Inputs:** The memory data bus inputs provide a separate input port for memory data.	Active low
23	ED	**Memory Data Enable Input:** When in the low state, the memory data enable input enables the memory data outputs ($D_0 - D_1$).	Active low
28	V_{CC}	**+5 Volt Supply**	

SYSTEM DESCRIPTION

Microfunction Decoder and K-Bus

Basic microfunctions are controlled by a 7-bit bus ($F_0 - F_6$) which is organized into 2 groups. The higher 3 bits ($F_4 - F_6$) are designated as F-Group and the lower 4 bits ($F_0 - F_3$) are designated as the R-Group. The F-Group specifies the type of operation to be performed and the R-Group specifies the registers involved.

The F-Bus instructs the microfunction decoder to:
- Select ALU functions to be performed
- Generate scratchpad register address
- Control A and B multiplexer

The resulting microfunction action can be:
- Data transfer
- Shift operations
- Increment and decrement
- Initialize stack
- Test for zero conditions
- 2's complement addition and subtraction
- Bit masking
- Maintain program counter

A and B Multiplexers

A and B multiplexers select the proper 2 operands to the ALU.

A multiplexer selects inputs from one of the following:
- M-bus (data from main memory)
- Scratchpad registers
- Accumulator

B multiplexer selects inputs from one of the following:
- I-bus (data from external I/O devices)
- Accumulator
- K-bus (literal or masking information from micro-program memory)

Scratchpad Registers

- Contains 11 registers ($R_0 - R_9$, T)
- Scratchpad register outputs are multiplexed to the ALU via the A multiplexer
- Used to store intermediate results from arithmetic/logic operations
- Can be used as program counter

Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations of the CPE.

Arithmetic operations are:
- 2's complement addition
- Incrementing
- Decrementing
- Shift left
- Shift right

Logical operations are:
- Transfer
- AND
- Inclusive-OR
- Exclusive-NOR
- Logic complement

ALU operation results are then stored in the accumulator and/or scratchpad registers. For easy expansion to larger arrays carry look-ahead outputs (X and Y) and cascading shift inputs (LI, RO) are provided.

Central Processing Element N3002

Accumulator
- Stores results from ALU operations
- The output of accumulator is multiplexed into ALU via the A and B multiplexer as one of the operands

Input Buses
M-bus: Data bus from main memory
- Accepts 2 bits of data from main memory into CPE
- Is multiplexed into the ALU via the A multiplexer

I-bus: Data bus from input/output devices Accepts 2 bits of data from external input/output devices into CPE
- Is multiplexed into the ALU via the B multiplexer

K-bus: A special feature of the N3002 CPE
- During arithmetic operations, the K-bus can be used to **mask** portions of the field being operated on
- Select or remove accumulator from operation by placing K-bus in all "1" or all "0" state respectively
- During non-arithmetic operation, the carry circuit can be used in conjunction with the K-bus for word-wise-OR operation for bit testing
- Supply literal or constant data to CPE

Output Buses
A-bus and Memory Address Register
- Main memory address is stored in the memory address register (MAR)
- Main memory is addressed via the A-bus
- MAR and A-bus may also be used to generate device address when executing I/O instructions
- A-bus has Tri-State outputs

D-bus: Data bus from CPE to main memory or to I/O devices
- Sends buffered accumulator outputs to main memory or the external I/O devices
- D-bus has Tri-State outputs

FUNCTION DESCRIPTION

F GROUP	R GROUP	K BUS	NAME	EQUATION	DESCRIPTION
0	I	XX	—	$R_n + (AC\ K) + CI \rightarrow R_n, AC$	**Logically AND AC with K-bus.** Add the result to R_n and carry input (CI). Deposit the sum in AC and R_n.
		OO	ILR	$R_n + CI \rightarrow R, AC$	Conditionally increment R_n and load the result in AC. Used to load AC from R_n or to increment R_n and load a copy of the results in AC.
		11	ALR	$AC + R_n + CI \rightarrow R_n, AC$	Add AC and CI to R_n and load the result in AC. Used to add AC to a register. If R_n is AC, then AC is shifted left one bit position.
0	II	XX	—	$M + (AC\ K) + CI \rightarrow AT$	**Logically AND AC with the K-bus.** Add the result to CI and the M-bus. Deposit the sum in AC or T.
		OO	ACM	$M + CI \rightarrow AT$	Add CI to M-bus. Load the result in AC or T, as specified. Used to load memory data in the specified register, or to load incremented memory data in the specified register.
		11	AMA	$M + AM + CI \rightarrow AT$	Add the M-bus to AC and CI, and load the result in AC or T, as specified. Used to add memory data or incremented memory data to AC and store the sum in the specified register.
0	III	XX	—	$AT_L \vee (\overline{I_L \wedge K_L}) \rightarrow RO$ $LI \vee [(I_H \wedge K_H) \wedge AT_H] \rightarrow AT_H$ $[AT_L \wedge (I_L \wedge K_L)]$ $[AT_H \vee (I_H \wedge K_H)] \rightarrow AT_L$	None
		OO	SRA	$AT_L \rightarrow RO$ $AT_H \rightarrow AT_L$ $L_I \rightarrow AT_H$	Shift AC or T, as specified, right one bit position. Place the previous low order bit value on RO and fill the high order bit from the data on LI. Used to shift or rotate AC or T right one bit.
1	I	XX	—	$K \vee R_n \rightarrow MAR$ $R_n + K + CI \rightarrow R_n$	**Logically OR R_n with the K-bus.** Deposit the result in MAR. Add the K-bus to R_n and CI. Deposit the result in R_n.
		OO	LMI	$R_n \rightarrow MAR, R_n + CI \rightarrow R_n$	Load MAR to R_n. Conditionally increment R_n. Used to maintain a macro-instruction program counter.
		11	DSM	$11 \rightarrow MAR, R_n - 1 + CI \rightarrow R_n$	Set MAR to all ones. Conditionally decrement R_n by one. Used to force MAR to its highest address and to decrement R_n.
1	II	XX	—	$K \vee M \rightarrow MAR$ $M + K + CI \rightarrow AT$	**Logically OR the M-bus with the K-bus.** Deposit the result in MAR. Add the K-bus to the M-bus and CI. Deposit the sum in AC or T.
		OO	LMM	$M \rightarrow MAR, M + CI \rightarrow AT$	Load MAR from the M-bus. Add CI to the M-bus. Deposit the result in AC or T. Used to load the address register with memory data for macro-instructions using indirect addressing.
		11	LDM	$11 \rightarrow MAR$ $M - 1 + CI \rightarrow AT$	Set MAR to all ones. Subtract one from the M-bus. Add CI to the difference and deposit the result in AC or T, as specified. Used to load decremented memory data in AC or T.

FUNCTION DESCRIPTION (Continued)

F GROUP	R GROUP	K BUS	NAME	EQUATION	DESCRIPTION
1	III	XX	—	$(\overline{AT} \vee K) + (AT \wedge K) + CI \rightarrow AT$	**Logically OR the K-bus with the complement of AC or T, as specified.** Add the result to the logical AND of specified register with the K-bus. Add the sum to CI. Deposit the result in the specified register.
		OO	CIA	$\overline{AT} + CI \rightarrow AT$	Add CI to the complement of AC or T, as specified. Deposit the result in the specified register. Used to form the 1's or 2's complement of AC or T.
		11	DCA	$\overline{AT} - 1 + CI \rightarrow AT$	Subtract one from AC or T, as specified. Add CI to the difference and deposit the sum in the specified register. Used to decrement AC or T.
2	I	XX	—	$(AC \wedge K) - 1 + CI \rightarrow R_n$	**Logically AND the K-bus with AC.** Subtract one from the result and add the difference to CI. Deposit the sum in R_n.
		OO	CSR	$CI - 1 \rightarrow R_n$ (See Note 1)	Subtract one from CI and deposit the difference in R_n. Used to conditionally clear or set R_n to all 0's or 1's, respectively.
		11	SDR	$AC - 1 + CI \rightarrow R_n$ (See Note 1)	Subtract one from AC and add the difference to CI. Deposit the sum in R_n. Used to store AC in R_n, or to store the decremented value of AC in R_n.
2	II	XX	—	$(AC \wedge K) - 1 + CI \rightarrow AT$ (See Note 1)	**Logically AND the K-bus with AC.** Subtract one from the result and add the difference to CI. Deposit the sum in AC or T, as specified.
		OO	CSA	$CI - 1 \rightarrow AT$ (See Note 1)	Subtract one from CI and deposit the difference in AC or T. Used to conditionally clear or set AC or T.
		11	SDA	$AC - 1 + CI \rightarrow AT$ (See Note 1)	Subtract one from AC and add the difference to CI. Deposit the sum in AC or T. Used to store AC in T, or decrement AC, or store the decremented value of AC in T.
2	III	XX	—	$(I \wedge K) - 1 + CI \rightarrow AT$ (See Note 1)	**Logically AND the data of the K-bus with the data on the I-bus.** Subtract one from the result and add the difference to CI. Deposit the sum in AC or T, as specified.
		OO	CSA	$CI - 1 \rightarrow AT$	Subtract one from CI and deposit the difference in AC or T. Used to conditionally clear or set AC or T.
		11	LDI	$I - 1 + CI \rightarrow AT$	Subtract one from the data on the I-bus and add the difference to CI. Deposit the sum in AC or T, as specified. Used to load input bus data or decremented input bus data in the specified register.
3	I	XX	—	$R_n + (AC \wedge K) + CI \rightarrow R_n$	**Logically AND AC with the K-bus.** Add R_n and CI to the result. Deposit the sum in R_n.
		OO	INR	$R_n + CI \rightarrow R_n$	Add CI to R_n and deposit the sum in R_n. Used to increment R_n.
		11	ADR	$AC + R_n + CI \rightarrow R_n$	Add AC to R_n. Add the result to CI and deposit the sum in R_n. Used to add the accumulator to a register or to add the incremented value of the accumulator to a register.
3	II	XX	—	$M + (AC \wedge K) + CI \rightarrow AT$	**Logically AND AC with the K-bus.** Add the result to CI and the M-bus. Deposit the sum in AC or T.
		OO	ACM	$M + CI \rightarrow AT$	Add CI to M-bus. Load the result in AC or T, as specified. Used to load memory data in the specified register, or to load incremented memory data in the specified register.
		11	AMA	$M + AC + CI \rightarrow AT$	Add the M-bus to AC and CI, and load the result in AC or T, as specified. Used to add memory data or incremented memory data to AC and store the sum in the specified register.

FUNCTION DESCRIPTION (Continued)

F GROUP	R GROUP	K BUS	NAME	EQUATION	DESCRIPTION
3	III	XX	—	AT + (I∧K) + CI → AT	**Logically AND the K-bus with the I-bus.** Add CI and the contents of AC or T, as specified, to the result. Deposit the sum in the specified register.
		OO	INA	AT + CI → AT	Conditionally increment AC or T. Used to increment AC or T.
		11	AIA	I + AT + CI → AT	Add the I-bus to AC or T. Add CI to the result and deposit the sum in the specified register. Used to add input data or incremented input data to the specified register.
4	I	XX	—	CI ∨ (R_n∧AC∧K) → CO R_n∧(AC∧K) → R_n	**Logically AND the K-bus with AC.** Logically AND the result with the contents of R_n. Deposit the final result in R_n. Logically OR the value of CI with the word-wise OR of the bits of the final result. Place the value of the carry OR on the carry output (CO) line.
		OO	CLR	CI → CO, O → R_n	Clear R_n to all O's. Force CO to CI. Used to clear a register and force CO to CI.
		11	ANM	CI ∨ (R_n∧AC) → CO R_n∧AC → R_n	Logically AND AC with R_n. Deposit the result in R_n. Force CO to one if the result is non-zero. Used to AND the accumulator with a register and test for a zero result.
4	II	XX	—	CI ∨ (M∧AC∧K) → CO M∧(AC∧K) → AT	**Logically AND the K-bus with AC.** Logically AND the result with the M-bus. Deposit the final result in AC or T. Logically OR the value of CI with the word-wise OR of the bits of the final result. Place the value of the carry OR on CO.
		OO	CLA	CI → CO, O → AT	Clear AC or T, as specified, to all O's. Force CO to CI. Used to clear the specified register and force CO to CI.
		11	ANM	CI ∨ (M∧AC) → CO M∧AC → AT	Logically AND the M-bus with AC. Deposit the result in AC or T. Force CO to one if the result is non-zero. Used to AND the M-bus data to the accumulator and test for a zero result.
4	III	XX	—	CI ∨ (AT∧1∧K) → CO AT∧(I∧K) → AT	**Logically AND the I-bus with the K-bus.** Logically AND the result with AC or T. Deposit the final result in the specified register. Logically OR CI with the word-wise OR of the final result. Place the value of the carry OR on CO.
		OO	CLA	CI → CO, O → AT	Clear AC or T, as specified, to all O's. Force CO to CI. Used to clear the specified register and force CO to CI.
		11	ANI	CI ∨ (AT∧I) → CO AT∧1 → AT	Logically AND the I-bus with AC or T, as specified. Deposit the result in the specified register. Force CO to one if the result is non-zero. Used to AND the I-bus to the accumulator and test for a zero result.
5	I	XX	—	CI ∨ (R_n∧K) → CO K∧R_n → R_n	**Logically AND the K-bus with R_n.** Deposit the result in R_n. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.
		OO	CLR	CI → CO, O → R_n	Clear R_n to all O's. Force CO to CI. Used to clear a register and force CO to CI.
		11	TZR	CI ∨ R_n → CO R_n → R_n	Force CO to one if R_n is non-zero. Used to test a register for zero. Also used to AND K-bus data with a register for masking and, optionally, testing for a zero result.
5	II	XX	—	CI ∨ (M∧K) → CO K∧M → AT	**Logically AND the K-bus with the M-bus.** Deposit the result in AC or T, as specified. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.
		OO	CLA	CI → CO, O → AT	Clear AC or T, as specified, to all O's. Force CO to CI. Used to clear the specified register and force CO to CI.
		11	LTM	CI ∨ M → CO M → AT	Load AC or T, as specified, from the M-bus. Force CO to one if the result is non-zero. Used to load the specified register from memory and test for zero result. Also used to AND the K-bus with the M-bus for masking and, optionally, testing for a zero result.

FUNCTION DESCRIPTION (Continued)

F GROUP	R GROUP	K BUS	NAME	EQUATION	DESCRIPTION
5	III	XX	—	$CI \vee (AT \wedge K) \rightarrow CO$ $K \wedge AT \rightarrow AT$	**Logically AND the K-bus with AC or T, as specified.** Deposit the result in the specified register. Logically OR CI with the word-wise OR of the result. Place the value of the carry OR on CO.
		OO	CLA	$CI \rightarrow CO$, $O \rightarrow AT$	Clear AC or T, as specified, to all O's. Force CO to CI. Used to clear the specified register and force CO to CI.
		11	TZA	$CI \vee AT \rightarrow CO$ $AT \rightarrow AT$	Force CO to one if AC or T, as specified, is non-zero. Used to test the specified register for zero. Also used to AND the K-bus to the specified register for masking and, optionally, testing for a zero result.
6	I	XX	—	$CI \vee (AC \wedge K) \rightarrow CO$ $R_n \vee (AC \wedge K) \rightarrow R_n$	**Logically OR CI with the word-wise OR of the logical AND of AC and the K-bus.** Place the result of the carry OR on CO. Logically OR R_n with the logical AND of AC and the K-bus. Deposit the result in R_n.
		OO	NOP	$CI \rightarrow CO$, $R_n \rightarrow R_n$	Force CO to CI. Used as a null operation or to force CO to CI.
		11	ORR	$CI \vee AC \rightarrow CO$ $R_n \vee AC \rightarrow R_n$	Force CO to one if AC is non-zero. Logically OR AC with R_n. Deposit the result in R_n. Used to OR the accumulator to a register and, optionally, test the previous accumulator value for zero.
6	II	XX	—	$CI \vee (AC \wedge K) \rightarrow CO$ $M \vee (AC \wedge K) \rightarrow AT$	**Logically OR CI with the word-wise OR of the logical AND of AC and the K-bus.** Place the carry OR on CO. Logically OR the M-bus, with the logical AND of AC and the K-bus. Deposit the final result in AC or T.
		OO	LMF	$CI \rightarrow CO$, $M \rightarrow AT$	Load AC or T, as specified, from the M-bus. Force CO to CI. Used to load the specified register with memory data and force CO to CI.
		11	ORM	$CI \vee AC \rightarrow CO$ $M \vee AC \rightarrow AT$	Force CO to one if AC is non-zero. Logically OR the M-bus with AC. Deposit the result in AC or T, as specified. Used to OR M-bus with the AC and, optionally, test the previous value of AC for zero.
6	III	XX	—	$CI \vee (I \wedge K) \rightarrow CO$ $AT \vee (I \wedge I) \rightarrow AT$	**Logical OR CI with the word-wise OR of the logical AND of the I-bus and the K-bus.** Place the carry OR on CO. Logically AND the K-bus with the I-bus. Logically OR the result with AC or T, as specified. Deposit the final result in the specified register.
		OO	NOP	$CI \rightarrow CO$, $AT \rightarrow AT$	Force CO to CI. Used as a null operation or to force CO to CI.
		11	ORI	$CI \vee I \rightarrow CO$ $I \vee AT \rightarrow$	Force CO to one if the data on the I-bus is non-zero. Logically OR the I-bus to AC to T, as specified. Deposit the result in the specified register. Used to OR I-bus data with the specified register and, optionally, test the I-bus data for zero.
7	I	XX	—	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO$ $R_n \overline{\oplus} (AC \vee K) \rightarrow R_n$	**Logically OR CI with the word-wise OR of the logical AND of R_n and AC and the K-bus.** Place the carry OR on CO. Logically AND the K-bus with AC. Exclusive-NOR the result with R_n. Deposit the final result in R_n.
		OO	CMR	$CI \rightarrow CO$, $R_n \rightarrow R_n$	Complement the contents of R_n. Force CO to CI.
		11	XNR	$CI\ (R_n \vee AC) \rightarrow CO$ $R_n \overline{\oplus} AC \rightarrow R_n$	Force CO to one if the logical AND of AC and R_n is non-zero. Exclusive-NOR AC with R_n. Deposit the result in R_n. Used to exclusive-NOR the accumulator with a register.

FUNCTION DESCRIPTION (Continued)

F GROUP	R GROUP	K BUS	NAME	EQUATION	DESCRIPTION
7	II	XX	—	$CI \vee (M \wedge AC \wedge K) \rightarrow CO$ $M \bar{\oplus} (AC \wedge K) \rightarrow AT$	**Logically OR CI with the word-wise OR of the logical AND of AC and the K-bus and M-bus.** Place the carry OR on CO. Logically AND the K-bus with AC. Exclusive NOR the result with the M-bus. Deposit the final result in AC or T.
		OO	LCM	$CI \rightarrow CO, \overline{M} \rightarrow AT$	**Load the complement of the M-bus into AC or T, as specified.** Force CO to CI.
		11	XNM	$CI\ (M \wedge AC) \rightarrow CO$ $M \bar{\oplus} AC \rightarrow AT$	**Force CO to one if the logical AND of AC and the M-bus is non-zero.** Exclusive-NOR AC with the M-bus. Deposit the result in AC or T, as specified. Used to exclusive-NOR memory data with the accumulator.
7	III	XX	—	$CI \vee (AT \wedge I \wedge K) \rightarrow CO$ $AT \bar{\oplus} (I \wedge K) \rightarrow AT$	**Logically OR CI with the word-wise OR of the logical AND of the specified register and the I-bus and K-bus.** Place the carry OR on CO. Logically AND the K-bus with the I-bus. Exclusive-NOR the result with AC or T, as specified. Deposit the final result in the specified register.
		OO	CMA	$CI \rightarrow CO\ \overline{AT} \rightarrow AT$	**Complement AC or T, as specified.** Force CO. to CI.
		11	XNI	$CI \vee (AT \wedge I) \rightarrow CO$ $I \bar{\oplus} AT \rightarrow AT$	**Force CO to one if the logical AND of the specified register and the I-bus is non-zero.** Exclusive-NOR AC with the I-bus. Deposit the result in AC or T, as specified. Used to exclusive-NOR input data with the accumulator.

NOTE:
2's complement arithmetic adds 111 ... 11 to perform subtraction of 000 ... 01.

FUNCTION DESCRIPTION KEY

SYMBOL	MEANING
I, K, M	Data on the I, K, and M buses, respectively
CI, LI	Data on the carry input and left input, respectively
CO, RO	Data on the carry output and right output, respectively
R_n	Contents of register n including T and AC (R-Group I)
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
MAR	Contents of the memory address register
L, H	As subscripts, designate low and high order bit, respectively
+	2's complement addition
−	2's complement subtraction
∧	Logical AND
∨	Logical OR
⊕	Exclusive-NOR
→	Deposit into

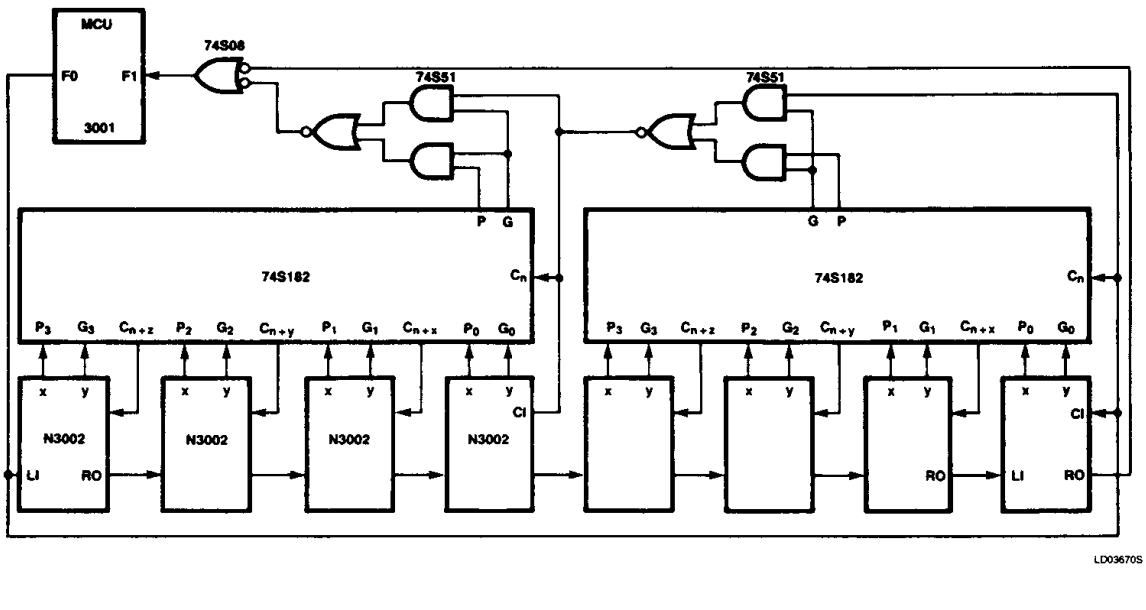
Central Processing Element N3002

AC ELECTRICAL CHARACTERISTICS N3001 = T_A = 0°C to +70°C, V_{CC} = 5V ±5%

PARAMETER		N3002			UNIT
		Min	Typ*	Max	
tCY	Clock cycle time	70	45		ns
tWP	Clock pulse width	17	10		ns
tFS	Function input set-up time (F_0 through F_6)	48	−23 → 35		ns
Data set-up time:					
tDS	I_0, I_1, M_0, M_1, K_0, K_1	40	12 → 29		ns
tSS	LI, CI	21	0 → 7		30
Data and function hold time:					
tFH	F_0 through F_6	4	0		ns
tDH	I_0, I_1, M_0, M_1, K_0, K_1	4	−28 → −11		ns
tSH	LI, CI	12	−7 → 0		ns
Propagation Delay to X, Y, RO from:					
tXF	Any Function Input		28	52	ns
tXD	Any Data Input		16 → 20	33	ns
tXT	Trailing Edge of CLK		33	48	ns
tXL	Leading Edge of CLK	13	18 → 40	70	ns
Propagation delay to CO from:					
tCL	Leading Edge of CLK	16	25 → 44	70	ns
tCL	Trailing Edge of CLK		30 → 40	56	ns
tCF	Any Function Input		25 → 35	52	ns
tCD	Any Data Input		17 → 23	55	ns
tCC	CI (Ripple Carry)		9 → 13	20	ns
Propagation delay to A_0, A_1, D_0, D_1 from:					
tDL	Leading Edge of CLK		17 → 25	40	ns
tDE	Enable Input ED, EA		10 → 12	20	ns

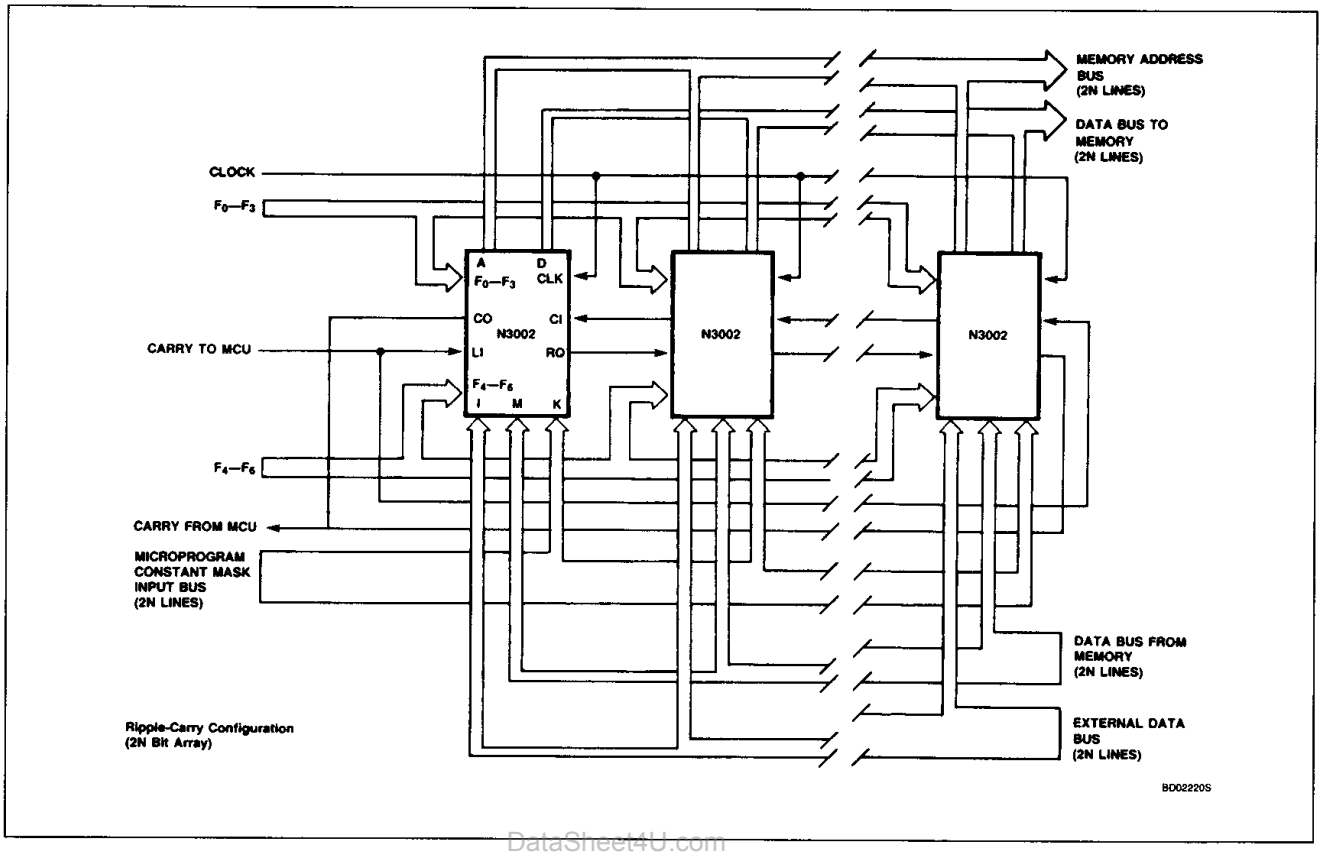
***NOTE:**
Typical values are for T_A = 25°C and typical supply voltage.

CARRY LOOK-AHEAD CONFIGURATION

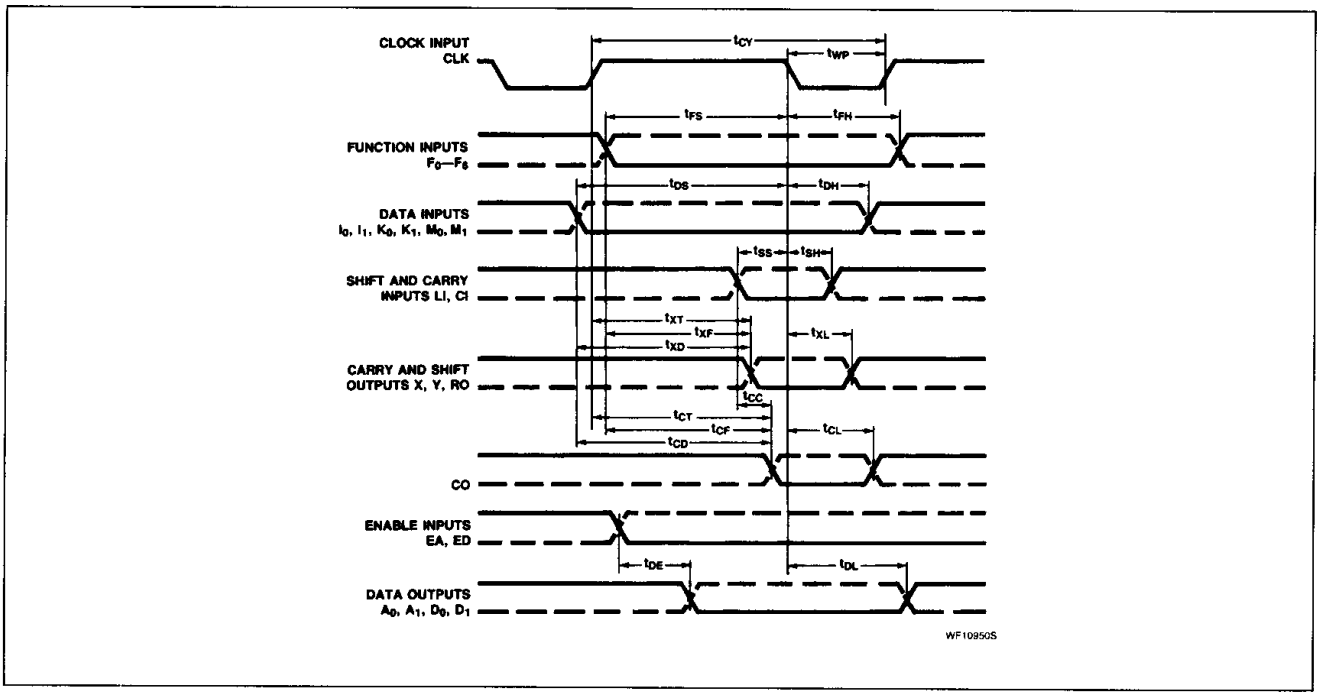


Central Processing Element N3002

TYPICAL CONFIGURATIONS



VOLTAGE WAVEFORMS



7