34706

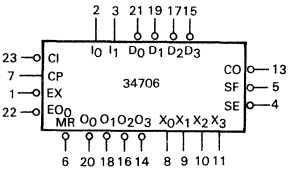
PROGRAM STACK

FAIRCHILD CMOS MACROLOGIC*

DESCRIPTION — The 34706 is a 16-word by 4-bit "Push-Down Pop-Up" Program Stack. It is designed to implement Program Counter (PC) and return address storage for nested subroutines in programmable digital systems. The 34706 executes four instructions: Return, Branch, Call and Fetch as specified by a 2-bit instruction. When the device is initialized, the program counter (PC) is in the top location of the stack. As a new PC value is "pushed" into the stack (Call Operation), all previous PC values effectively move down one level. The top location of the stack is the current PC. Up to 16 PC values can be stored, which gives the 34706 a 15 level nesting capability. "Popping" the stack (Return Operation) brings the most recent PC to the top of the stack and makes it available at the two output buses. The remaining two instructions affect only the top location of the stack. In the Branch Operation a new PC value is loaded into the top location of the stack from the $\overline{D}_0 - \overline{D}_3$ inputs. In the Fetch Operation, the contents of the top stack location (current PC value) are put on the $X_0 - X_3$ bus and the current PC value is incremented.

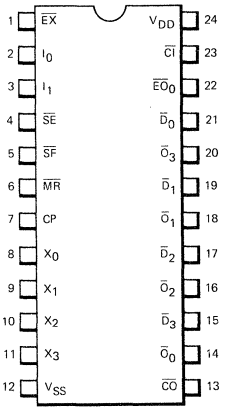
The 34706 may be expanded to any word length without additional logic. Three-state output drivers are provided on the 4-bit Address ($X_0 - X_3$) and Data Outputs, ($\overline{O}_0 - \overline{O}_3$); the X-bus outputs are enabled internally and only during the Fetch Instruction whereas the O-bus outputs are controlled by an Output Enable ($\overline{EO}_0$). Two status outputs, Stack Full ($\overline{SF}$) and Stack Empty ($\overline{SE}$) are provided. The 34706 is a member of Fairchild's 34000 CMOS Macrologic family, and is available in the new slim 24-pin package.

LOGIC SYMBOL



V_{DD} = Pin 24
V_{SS} = Pin 12

CONNECTION DIAGRAM
DIP (TOP VIEW)



- **16-WORD BY 4-BIT LIFO**
- **15-LEVEL NESTING CAPABILITY**
- **VERY LOW POWER — IDEAL FOR BATTERY OPERATION**
- **RELATIVE ADDRESSING CAPABILITY**
- **2 MHz MICROINSTRUCTION RATE**
- **PROGRAM COUNTER LOADABLE FROM DATA BUS**
- **OPTIONAL AUTOMATIC INCREMENT OF PROGRAM COUNTER**
- **STACK LIMIT STATUS INDICATORS**
- **NEW SLIM 24-PIN DIP**

PIN NAMES

$\overline{D}_0$-$\overline{D}_3$	Data Inputs (Active LOW)
I_0, I_1	Instruction Inputs
$\overline{EX}$	Execute Input (Active LOW)
CP	Clock Input
$\overline{MR}$	Master Reset Input (Active LOW)
$\overline{CI}$	Carry Input (Active LOW)
$\overline{EO}_0$	Output Enable Input (Active LOW)
$\overline{O}_0$-$\overline{O}_3$	Output Data Outputs (Active LOW)
$\overline{X}_0$-$\overline{X}_3$	Address Outputs
$\overline{CO}$	Carry Output (Active LOW)
$\overline{SF}$	Stack Full Output (Active LOW)
$\overline{SE}$	Stack Empty Output (Active LOW)

*A Trademark of Fairchild Camera and Instrument Corporation

FUNCTIONAL DESCRIPTION – As shown in the Block Diagram, the 34706 consists of an input multiplexer, a 16 x 4 RAM with output latches addressed by the Stack Pointer (SP), an incrementor, control logic, and output buffers. The 34706 is organized around three 4-bit buses; the Input Data (D) Bus ($\overline{D}_0$, $\overline{D}_1$, $\overline{D}_2$, $\overline{D}_3$), Output Data (O) Bus ($\overline{O}_0$, $\overline{O}_1$, $\overline{O}_2$, $\overline{O}_3$) and the Address (X) Bus (X_0, X_1, X_2, X_3). The 34706 implements four instructions as determined by inputs I_0 and I_1. (See Table I). The O-bus is derived from the RAM output latches and enabled by the active LOW Output Enable ($\overline{EO}_0$) input. The X-bus is also derived from the output latches; it is enabled internally during the Fetch Instruction. Execution of instructions is controlled by the Execute ($\overline{EX}$) and Clock (CP) inputs.

FETCH OPERATION – The Fetch Operation places the content of the current Program Counter (PC) on the X-bus. If the Carry In ($\overline{CI}$) is LOW, the current PC is incremented in preparation for the next Fetch. If $\overline{CI}$ is HIGH, the value of the current program is unchanged, (Iterative Fetch).

The instruction code is set up on the I lines when CP is HIGH. The active level LOW Execute ($\overline{EX}$) is normally set up at this time as well. The control logic interprets I_0 and I_1 and selects the incrementor output as the data source to the RAM via the input multiplexer. The current PC value is loaded into the latches and is available on the O-bus if $\overline{EO}_0$ is LOW. When CP is LOW (assuming $\overline{EX}$ is also LOW) the output latches are disabled from following the RAM output and the X-bus Output buffers are enabled, applying the current PC to the X-bus. The output of the incrementor is written into the RAM during the period when CP and $\overline{EX}$ are LOW. If $\overline{CI}$ is LOW, the value stored in the current PC, plus one, is written into the RAM. If $\overline{CI}$ is HIGH, the current PC is not incremented. Carry Out ($\overline{CO}$) is LOW when the contents of the current PC is at its maximum, i.e., all ones. When CP or $\overline{EX}$ goes HIGH, writing into the RAM is inhibited and the Address buffers ($X_0 - X_3$) are disabled.

BRANCH OPERATION – During a Branch Operation, the Data Inputs ($D_0 - D_3$) are loaded into the current program counter.

The instruction code and the $\overline{EX}$ input are set up when CP is HIGH. The stack pointer remains unchanged. When CP goes LOW (assuming $\overline{EX}$ is LOW), the D-bus inputs are written into the current PC. The X-bus drivers are not enabled during a Branch Operation.

CALL OPERATION – During a Call Operation the content of the data bus is loaded into the top location of the stack and all previous PC values are effectively moved down one level.

The instruction code and the $\overline{EX}$ input are set up when CP is HIGH. When $\overline{EX}$ is LOW, a "one" is added to the stack pointer value thus incrementing the RAM address. When CP is LOW (assuming $\overline{EX}$ is LOW), the D-bus inputs are written into this new RAM location. On the LOW-to-HIGH CP transition, the incremented stack pointer value is loaded into the stack pointer register. When the RAM address is "1111" the Stack Full output ($\overline{SF}$) is LOW, indicating that no further Call Operations should be initiated. If an additional Call Operation is performed $\overline{SP}$ is incremented to "0000", the contents of that location will be written over, $\overline{SF}$ will go HIGH and the Stack Empty ($\overline{SE}$) will go LOW.

The X-bus drivers are not enabled during a Call Operation.

RETURN OPERATION – During the Return Operation the previous PC is "popped" to become the current PC.

The instruction is set up when CP is HIGH. When $\overline{EX}$ is LOW, a "one" is subtracted from the stack pointer value, thus decrementing the RAM address, presenting the "popped" PC value through the enabled latches to the three-state O-bus drivers. When CP is LOW, the latches are disabled, thereby holding the new current value of the PC. On the LOW-to-HIGH CP transition the decremented stack pointer value is loaded into the stack pointer register.

The X-bus drivers are not enabled during a return operation.

When the RAM address is "0000", the Stack Empty output ($\overline{SE}$) is LOW, indicating that no further return operations should be initiated. If an additional Return Operation is performed, $\overline{SP}$ is decremented to "1111", the $\overline{SE}$ will go HIGH and the Stack Full output ($\overline{SF}$) will go LOW. Operation of the active LOW Master Reset ($\overline{MR}$) causes the $\overline{SP}$ to be reset and the contents of that RAM location (0000) to be cleared. The Stack Empty ($\overline{SE}$) output goes LOW. This operation overrides all other inputs.

MULTIPLE 34706 OPERATION – The 34706 may be expanded to any word length in multiples of four without external logic. The connection for expanded operation is shown in Figure 1. Carry In ($\overline{CI}$) and Carry Out ($\overline{CO}$) are connected to provide automatic increment of the current program counter during the Fetch Operation. The $\overline{CI}$ input of the least significant 34706 is tied LOW to ground; the $\overline{CO}$ input of the least significant 34706 is connected to the $\overline{CI}$ input of the next significant 34706.

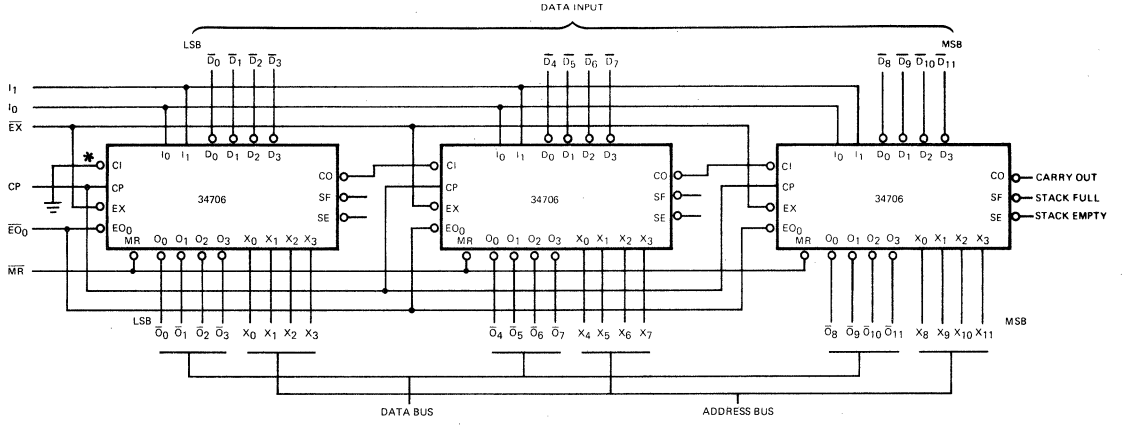


FIGURE 1. 34706 EXPANSION A 16 BY 12 PROGRAM STACK

*Tie to V_{DD} to disable automatic increment.

4

TABLE I
INSTRUCTION SET FOR THE 34706

I_1	I_0	INSTRUCTION	INTERNAL OPERATION	X-BUS	O-BUS (WITH $\overline{EO}_0$ LOW)
L	L	Return (Pop)	Decrement Stack Pointer	Disabled	New ("popped") Program Counter value when $\overline{EX}$ goes LOW
L	H	Branch (Load PC)	Load D-Bus into current Program Counter location	Disabled	Current Program Counter until CP goes HIGH again, then updated with newly entered PC value
H	L	Call (Push)	Increment Stack Pointer & Load D-Bus into new Program Counter Location	Disabled	Current Program Counter until CP goes HIGH again, then updated with newly entered PC value
H	H	Fetch (Increment PC)	Increment Current Program Counter if $\overline{CI}$ is LOW	Current Program Counter while both CP & $\overline{EX}$ are LOW, disabled while CP or $\overline{EX}$ is HIGH	Current Program Counter until CP goes HIGH again, then updated with incremented PC value

H = HIGH Level
L = LOW Level

34706 BLOCK DIAGRAM