Integrated Device Technology, Inc.

16-BIT CMOS MICROPROGRAM SEQUENCER

IDT49C410
IDT49C410A

FEATURES:

- 16-bit wide address path
  - — Address up to 65,536 words of microprogram memory
- 16-bit loop counter
  - — Pre-settable down-counter for counting loop iterations and repeating instructions
- Low-power CMOS
  - — Icc (max.)
    Military: 90mA
    Commercial: 75 mA
- Fast
  - — IDT49C410 meets 2910A speeds
  - — IDT49C410A is a 30% speed upgrade
- 33-deep stack
  - — Accommodates highly nested microcode
- 16 powerful microinstructions
- Available in 48-pin, 600 mil plastic and sidebraze DIP, 52-pin PLCC and 48-pin Flatpack
- Three enables control branch address sources
- Four address sources
- 2910A instruction compatibility
- Military product available compliant to MIL-STD-883, Class B
- Standard Military Drawing #5962-88643 is listed for this function

DESCRIPTION:

The IDT49C410s are architecture and function code compatible to the 2910A with an expanded 16-bit address path, thus allowing for programs up to 65,536 words in length. They are microprogram address sequencers intended for controlling the sequence of execution of microinstructions stored in the microprogram memory. Besides the capability of sequential access, they provide conditional branching to any microinstruction within their 65,536 microword range.

The 33-deep stack provides microsubroutine return linkage and looping capability. The deep stack can be used for highly nested microcode applications. Microinstruction loop count control is provided with a count capability of 65,536.

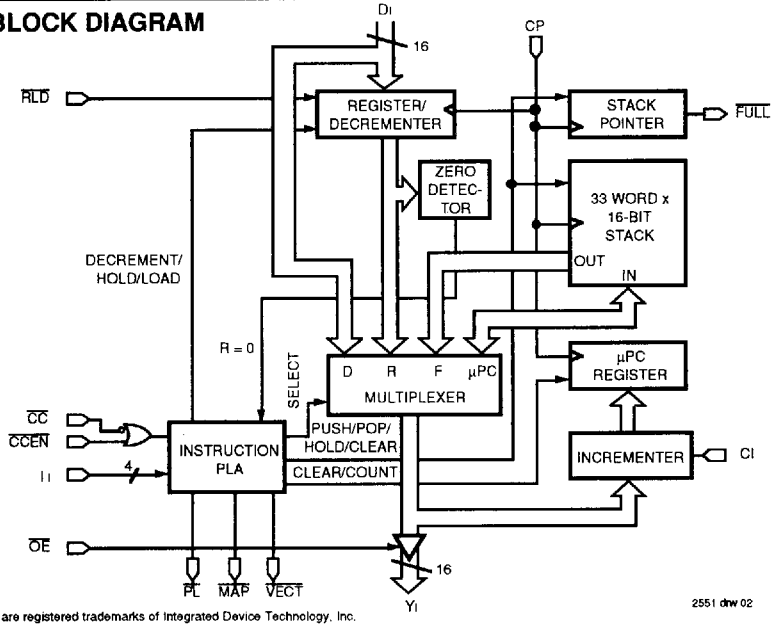
During each microinstruction, the microprogram controller provides a 16-bit address from one of four sources: 1) the microprogram address register (μPC), which usually contains an address one greater than the previous address; 2) an external (direct) input (D); 3) a register/counter (R) retaining data loaded during a previous microinstruction; or 4) a last-in/first-out stack (F).

The IDT49C10s are fabricated using CEMOS, a CMOS technology designed for high-performance and high-reliability.

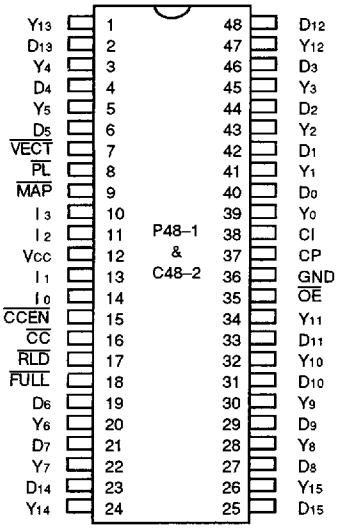
The IDT49C410s are pin-compatible, performance-enhanced, easily upgradable versions of the 2910A.

The IDT49C410s are available in 48-pin DIP (600 mil x 100 mil centers), 52-pin PLCC and 48-pin flatpack.

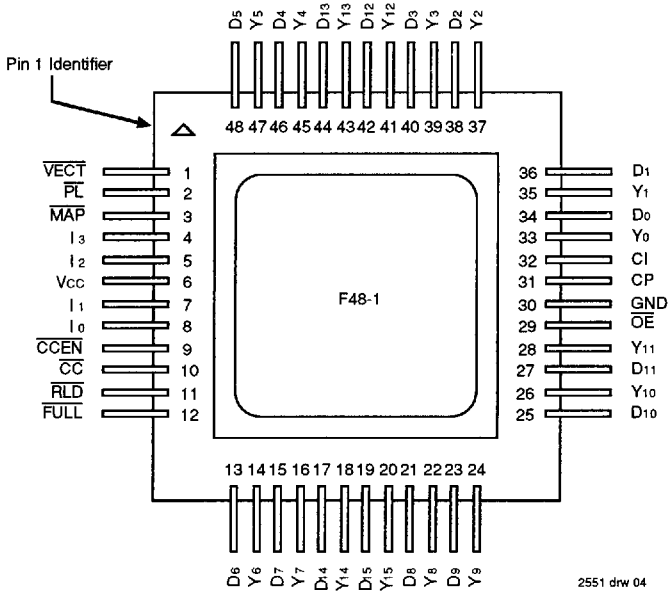
FUNCTIONAL BLOCK DIAGRAM



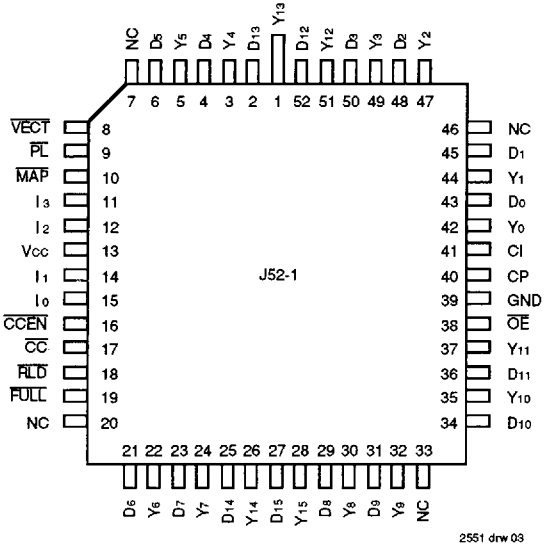
PIN CONFIGURATIONS



DIP
TOP VIEW



FLATPACK
TOP VIEW



PLCC
TOP VIEW

IDT49C410 PIN DESCRIPTIONS

Pin Name	I/O	Description
DI	I	Direct input to register/counter and multiplexer D0 is LSB.
II	I	Selects one-of-sixteen instructions.
$\overline{CC}$	I	Used as test criterion. A LOW on $\overline{CC}$ indicates "passed" test condition.
$\overline{CCEN}$	I	Whenever the signal is HIGH, $\overline{CC}$ is ignored and the device operates as though $\overline{CC}$ were true (LOW).
CI	I	Low order carry input to incrementer for microprogram counter.
$\overline{RLD}$	I	When LOW forces loading of register/ counter regardless of instruction or condition.
$\overline{OE}$	I	Three-state control of YI outputs.
CP	I	Triggers all internal state changes at LOW-to-HIGH edge.
YI	O	Address to microprogram memory. Y0 is LSB, Y15 is MSB.
$\overline{FULL}$	O	Indicates that 33 items are on the stack.
$\overline{PL}$	O	Can select #1 source (usually Pipeline Register) as direct input source.
$\overline{MAP}$	O	Can select #2 source (usually Mapping PROM or PLA) as direct input source.
$\overline{VECT}$	O	Can select #3 source (for example, Interrupt Starting Address) as direct input source.

2551 tbl 01

PRODUCT DESCRIPTION

The IDT49C410s are high-performance CMOS microprogram sequencers that are intended for use in very high-speed microprogrammable microprocessor applications. The sequencers allow for direct control of up to 64K words of microprogram.

The heart of the microprogram sequencers is a 4-input multiplexer that is used to select one of four address sources to select the next microprogram address. These address sources include the register/counter, the direct input, the microprogram counter or the stack as the source for the address of the next microinstruction.

The register/counter consists of sixteen D-type flip-flops which can contain either an address or a count. These edge-triggered flip-flops are under the control of a common clock enable, as well as the four microinstruction control inputs. When the load control ($\overline{RDL}$) is LOW, the data at the D inputs is loaded into this register on the LOW-to-HIGH transition of the clock. The output of the register/counter is available at the multiplexer as a possible next address source for the microcode. Also, the terminal count output associated with the register/counter is available at the internal instruction PLA to be used as condition code input for some of the microinstructions. The IDT49C410s contain a microprogram counter that usually contains the address of the next microinstruction compared to that currently being executed. The microprogram counter actually consists of a 16-bit incrementer followed by a 16-bit register. The microprogram counter will increment the address coming out of the sequencer going to the microprogram memory if the carry-in input to this counter is HIGH; otherwise, this address will be loaded into the microprogram counter. Normally, this carry-in input is set to the logic HIGH state so that the incrementer will be active. Should the carry-in input be set LOW, the same address is loaded into the microprogram counter. This is a technique that can be used to allow execution of the same microinstruction several times.

There are sixteen D-inputs on the IDT49C410s that go directly to the address multiplexer. These inputs are used to provide a branch address that can come directly from the microcode or some other external source. The fourth input available to the multiplexer for next address control is the 33-deep, 16-bit wide LIFO stack. The LIFO stack provides return address linkage for subroutines and loops. The IDT49C410s contain a built-in stack pointer that always points to the last stack location written. This allows for stack reference operations, usually called loops, to be performed without popping the stack.

The stack pointer internal to the IDT49C410s is actually an up/down counter. During the execution of microinstructions one, four and five, the PUSH operation may occur depending on the state of the condition code input. This causes the stack pointer to be incremented by one and the stack to be written with the required return linkage (the value contained in the microprogram counter). On the microprogram cycle following the PUSH, this new return linkage data that was in the microprogram counter is now at the new location pointed to by the stack pointer. Thus, any time the multiplexer looks at the stack, it will see this data on the top of the stack.

During five different microinstructions, a pop operation associated with the stack may occur. If the pop occurs, the stack pointer is decremented at the next LOW-to-HIGH transition of the clock. A pop decrements the stack pointer which is the equivalent of removing the old information from the top of the stack.

The IDT49C410s are designed so that the stack pointer linkage allows any sequence of pushes, pops or stack references to be used. The depth of the stack can grow to a full 33 locations. After a depth of 33 is reached, the $\overline{FULL}$ output goes LOW. If further PUSHes are attempted when the stack is full, the stack information at the top of the stack will be destroyed but the stack pointer will not end around. It is necessary to initialize the stack pointer when power is first turned on. This is performed by executing a RESET instruction (Instruction 0). This sets the stack pointer to the stack empty position — the equivalent depth of zero. Similarly, a pop from an empty stack may place unknown data on the Y outputs, but the stack pointer is designed not to end around. Thus, the stack pointer will remain at the 0 or stack empty location if a pop is executed while the stack is already empty.

The IDT49C410's internal 16-bit register/counter is used during microinstructions eight, nine and fifteen. During these instructions, the 16-bit counter acts as a down counter and the terminal count (count = 0) is used by the internal instruction PLA as an input to control the microinstruction branch test capability. The design of the internal counter is such that, if it is preloaded with a number N and then this counter is used in a microprogram loop, the actual sequence in the loop will be executed N + 1 times. Thus, it is possible to load the counter with a count of 0 and this will result in the microcode being executed one time. The 3-way branch microinstruction, instruction 15, uses both the loop counter and the external condition code input to control the final source address from the Y outputs of the microprogram sequencer. This 3-way branch may result in the next address coming from the D inputs, the stack or the microprogram counter.

The IDT49C410s provide a 16-bit address at the Y outputs that are under control of the $\overline{OE}$ input. Thus, the outputs can be put in the three-state mode, allowing the writable control store to be loaded or certain types of external diagnostics to be executed.

In summary, the IDT49C410s are the most powerful microprogram sequencers currently available. They provide the deepest stack, the highest performance and lowest power dissipation for today's microprogrammed machine design.

10

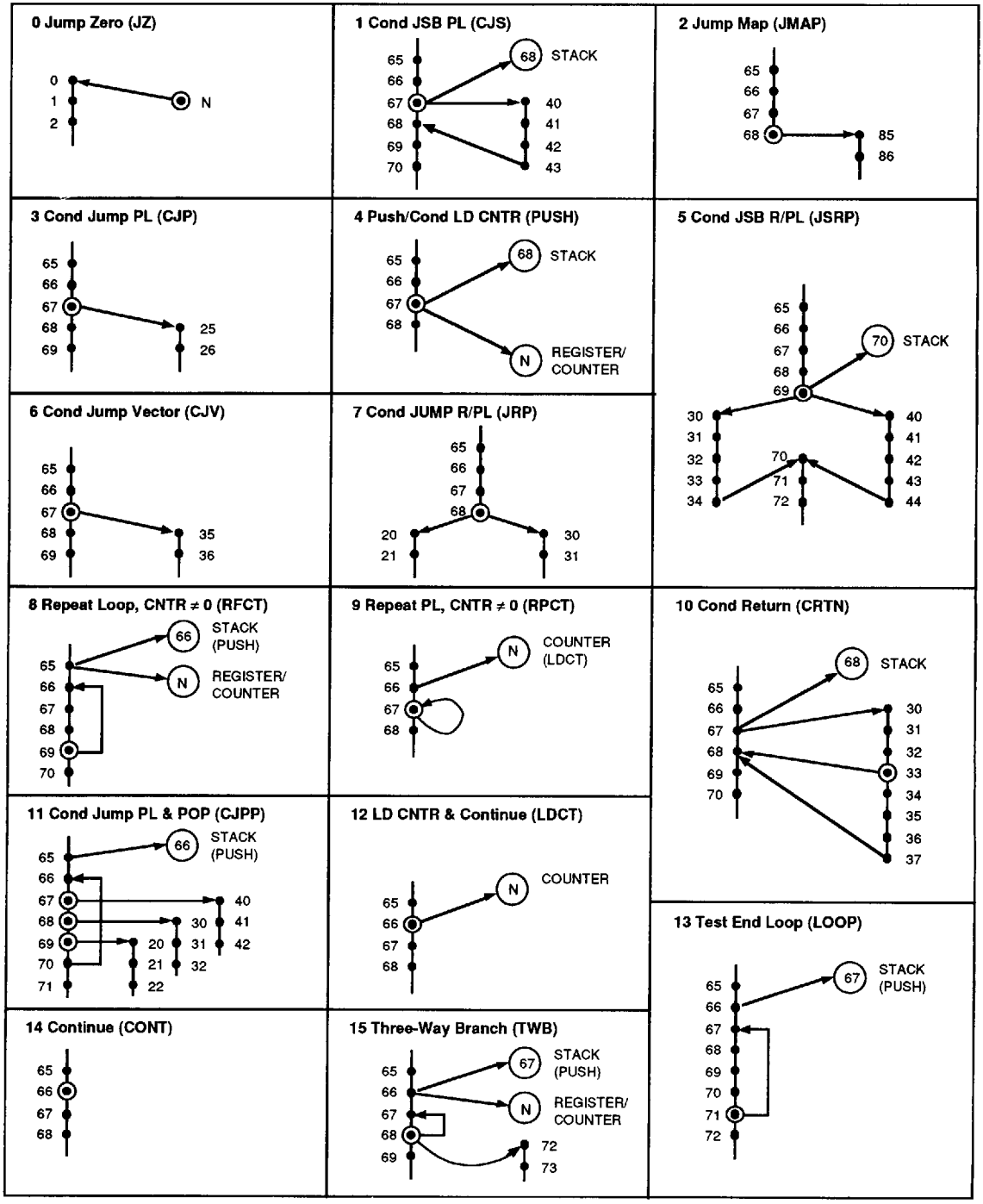


Figure 1. IDT49C410 Flow Diagrams

2551 drw 05

IDT49C410 OPERATION

The IDT49C410s are CMOS pin-compatible implementations of the Am2910 and 2910A microprogram sequencers. The IDT49C410 sequencers are functionally identical except that they are 16 bits wide and provide a 33-deep stack to give the microprogrammer more capability in terms of microprogram subroutines and microprogram loops. The definition of each microprogram instruction is shown in the table of instructions. This table shows the results of each instruction in terms of controlling the multiplexer, which determines the Y outputs, and in controlling the signals that can be used to enable various branch address sources ($\overline{PL}$, $\overline{MAP}$, $\overline{VECT}$). The operation of the register/counter and the 33-deep stack after the next LOW-to-HIGH transition of the clock are also shown. The internal multiplexer is used to select which of the internal sources is used to drive the Y outputs. The actual value loaded into the microprogram counter is either identical to the Y output or the Y output value is incremented by 1 and placed in the microprogram counter. This function is under the control of the carry inputs. For each of the microinstruction inputs, only one of the three outputs ($\overline{PL}$, $\overline{MAP}$ or $\overline{VECT}$) will be LOW. Note that this function is not determined by any of the possible condition code inputs. These outputs can be used to control the three-state selection of one of the sources for the microprogram branches.

Two inputs, $\overline{CC}$ and $\overline{CCEN}$, can be used to control the conditional instructions. These are fully defined in the table of instructions. The $\overline{RLD}$ input can be used to load the internal register/counter at any time. When this input is LOW, the data at the D inputs will be loaded into this register/counter on the LOW-to-HIGH transition of the clock. Thus, the $\overline{RLD}$ input overrides the internal hold or decrement operations specified by the various microinstructions. The $\overline{OE}$ input is normally LOW and is used as the three-state enable for the Y outputs. The internal stack in the IDT49C410s is a last-in/first-out memory that is 16-bits in width and 33 words deep. It has a stack pointer that addresses the stack and always points to the value currently on the top of the stack. When instruction 0 (RESET) is executed, the stack pointer is initialized to the top of the stack which is, by definition, the stack empty condition. Thus, the contents of the top of the stack are undefined until the forced PUSH occurs. A pop performed while the stack is empty will not change the stack pointer in any way; however, it will result in unknown data at the Y outputs.

By definition, the stack is full any time 33 more PUSHes than pops have occurred since the stack was last empty. When this happens, the $\overline{FULL}$ Flag will go LOW. This signal first goes LOW on the microcycle after the 33 pushes occur. When this signal is LOW, no additional pushes should be attempted or the information on the top of the stack will be lost.

THE IDT49C410 INSTRUCTION SET

This data sheet contains a block diagram of the IDT49C410 microprogram sequencers. As can be seen, the devices are controlled by a 4-bit microinstruction word (I_3 - I_0). Normally, this word is supplied from one 4-bit field of the microinstruction word associated with the entire state machine system. These four bits provide for the selection of one of the sixteen powerful instructions associated with selecting the address of the next microinstruction. Unused Y outputs can be left open; however, the corresponding most significant D inputs should be tied to ground for smaller microwords. This is necessary to make sure the internal operation of the counter is proper should less than 64K of microcode be implemented. As shown in the block diagram, the internal instruction PLA uses the four instruction inputs as well as the $\overline{CC}$, $\overline{CCEN}$ and the internal counter = 0 line for controlling the sequencer. This internal instruction PLA provides all of the necessary internal control signals to control each particular part of the microprogram sequencer. The next address at the Y outputs of the IDT49C410s can be from one of four sources. These include the internal microprogram counter, the last-in/first-out stack, the register/counter and the direct inputs.

The following paragraphs will describe each instruction associated with the IDT49C410s. As a part of the discussion, an example of each instruction is shown in Figure 1. The purpose of the examples is to show microprogram flow. Thus, in each example the microinstruction currently being executed has a circle around it. That is, this microinstruction is assumed to be the contents of the pipeline register at the output of the microprogram memory. In these drawings, each of the dots refers to the time that the contents of the microprogram memory word would be in the pipeline register and is currently being executed.

INSTRUCTION 0 – JUMP 0 (JZ)

This Conditional Jump is used at power-up time or at any restart sequence when the need is to reset the stack pointer and jump to the very first address in microprogram memory. The Jump 0 instruction does not change the contents of the register/counter.

INSTRUCTION 1 – CONDITIONAL JUMP TO SUBROUTINE (CJS)

The Conditional Jump to Subroutine instruction is the one used to call microprogram subroutines. The subroutine address will be contained in the pipeline register and presented at the D inputs. If the condition code test is passed, a branch is taken to the subroutine. Referring to the flow diagram for the IDT49C410s shown in Figure 1, we see that the content of the microprogram counter is 68. This value is pushed onto the stack and the top of stack pointer is incremented. If the test is failed, this Conditional Jump to Subroutine instruction behaves as a simple continue. That is, the content of microinstruction address 68 is executed next.

INSTRUCTION 2 – JUMP MAP (JMAP)

This sequencer instruction can be used to start different microprogram routines based on the machine instruction opcode. This is typically accomplished by using a mapping PROM as an input to the D inputs on the microprogram sequencer. The JMAP instruction branches to the address appearing on the D inputs. In the flow diagram shown in Figure 1, we see that the branch actually will be the contents of microinstruction 85 and this instruction will be executed next.

10

IDT49C410 INSTRUCTION OPERATIONAL SUMMARY

$I_3 - I_0$	Mnemonic	CC	Counter Test	Stack	Address Source	Register Counter	Enable Select
0	JZ	X	X	CLEAR	0	NC	$\overline{PL}$
1	CJS	PASS FAIL	X X	PUSH NC	D PC	NC NC	$\overline{PL}$ $\overline{PL}$
2	JMAP	X	X	NC	D	NC	$\overline{MAP}$
3	CJP	PASS FAIL	X X	NC NC	D PC	NC NC	$\overline{PL}$ $\overline{PL}$
4	PUSH	PASS FAIL	X X	PUSH PUSH	PC PC	LOAD NC	$\overline{PL}$ $\overline{PL}$
5	JSRP	PASS FAIL	X X	PUSH PUSH	D R	NC NC	$\overline{PL}$ $\overline{PL}$
6	CJV	PASS FAIL	X X	NC NC	D PC	NC NC	$\overline{VECT}$ $\overline{VECT}$
7	JRP	PASS FAIL	X X	NC NC	D R	NC NC	$\overline{PL}$ $\overline{PL}$
8	RFCT	X X	= 0 NOT = 0	POP NC	PC STACK	NC DEC	$\overline{PL}$ $\overline{PL}$
9	RPCT	X X	= 0 NOT = 0	NC NC	PC D	NC DEC	$\overline{PL}$ $\overline{PL}$
10	CRTN	PASS FAIL	X X	POP NC	STACK PC	NC NC	$\overline{PL}$ $\overline{PL}$
11	CJPP	PASS FAIL	X X	POP NC	D PC	NC NC	$\overline{PL}$ $\overline{PL}$
12	LDCT	X	X	NC	PC	LOAD	$\overline{PL}$
13	LOOP	PASS FAIL	X X	POP NC	PC STACK	NC NC	$\overline{PL}$ $\overline{PL}$
14	CONT	X	X	NC	PC	NC	$\overline{PL}$
15	TWB	PASS PASS FAIL FAIL	= 0 NOT = 0 = 0 NOT = 0	POP POP POP NC	PC PC D STACK	NC DEC NC DEC	$\overline{PL}$ $\overline{PL}$ $\overline{PL}$ $\overline{PL}$

NC = No Change; DEC = Decrement

2551 Tbl 02

INSTRUCTION 3 – CONDITIONAL JUMP PIPELINE (CJP)

The simplest branching control available in the IDT49C410 microprogram sequencers is that of conditional jump to address. In this instruction, the jump address is usually contained in the microinstruction pipeline register and presented to the D inputs. If the test is passed, the jump is taken while, if the test fails, this instruction executes as a simple continue. In the example shown in the flow diagram of Figure 1, we see that if the test is passed, the next microinstruction to be executed is the content of address 25. If the test is failed, the microcode simply continues to the contents of the next instruction.

INSTRUCTION 4 – PUSH/CONDITIONAL LOAD COUNTER (PUSH)

With this instruction, the counter can be conditionally loaded during the same instruction that pushes the current value of the microprogram counter on to the stack. Under any condition independent of the conditional testing, the microprogram counter is pushed on to the stack. If the conditional test is passed, the counter will be loaded with the value on the D inputs to the sequencer. If the test fails, the contents of the counter will not change. The PUSH/Conditional Load Counter instruction is used in conjunction with the loop instruction (Instruction 13), the repeat file based on the counter instruction (Instruction 9) or the 3-way branch instruction (Instruction 15).

INSTRUCTION 5 – CONDITIONAL JUMP TO SUBROUTINE R/PL (JSRP)

Subroutines may be called by a Conditional Jump Subroutine from the internal register or from the external pipeline register. In this instruction the contents of the microprogram counter are pushed on the stack and the branch address for the subroutine call will be taken from either the internal register/counter or the external pipeline register presented to the D inputs. If the conditional test is passed, the subroutine address will be taken from the pipeline register. If the conditional test fails, the branch address is taken from the internal register/counter. An example of this is shown in the flow diagram of Figure 1.

INSTRUCTION 6 – CONDITIONAL JUMP VECTOR (CJV)

The Conditional Jump Vector instruction is similar to the Jump Map instruction in that it allows a branch operation to a microinstruction as defined from some external source,

except that it is conditional. The Jump Map instruction is unconditional. If the conditional test is passed, the branch is taken to the new address on the D inputs. If the conditional test is failed, no branch is taken but rather the microcode simply continues to the next sequential microinstruction. When this instruction is executed, the $\overline{VECT}$ output is LOW unconditionally. Thus, an external 12-bit field can be enabled on to the D inputs of the microprogram sequencer.

INSTRUCTION 7 –
CONDITIONAL JUMP R/PL (JRP)

The Conditional Jump register/counter or external pipeline register always causes a branch in microcode. This jump will be to one of two different locations in the microcode address space. If the test is passed, the jump will be to the address presented on the D inputs to the microprogram sequencer. If the conditional test fails, the branch will be to the address contained in the internal register/counter.

INSTRUCTION 8 –
REPEAT LOOP COUNTER NOT EQUAL TO 0 (RFCT)

This instruction utilizes the loop counter and the stack to implement microprogrammed loops. The start address for the loop would be initialized by using the PUSH/Conditional Load Counter instruction. Then, when the repeat loop instruction is executed, if the counter is not equal to 0, the next microword address will be taken from the stack. This will cause a loop to be executed as shown in the Figure 1 flow diagram. Each time the microcode sequence goes around the loop, the counter is decremented. When the counter reaches 0, the stack will be popped and the microinstruction address will be taken from the microprogram counter. This instruction performs a timed wait or allows a single sequence to be executed the desired number of times. Remember, the actual number of loops performed is equal to the value in the counter plus 1.

INSTRUCTION 9 –
REPEAT PIPELINE, COUNTER NOT EQUAL TO 0 (RPCT)

This instruction is another technique for implementing a loop using the counter. Here, the branch address for the loop is contained in the pipeline register. This instruction does not use the stack in any way as a part of its implementation. As long as the counter is not equal to 0, the next microword address will be taken from the D inputs of the microprogram sequencer. When the counter reaches 0, the internal multiplexer will select the address source from the microprogram counter, thus causing the microcode to continue on and leave the loop.

INSTRUCTION 10 –
CONDITIONAL RETURN (CRTN)

The Conditional Return instruction is used for terminating subroutines. The fact that it is conditional allows the subroutine either to be ended or to continue. If the conditional test is passed, the address of the next microinstruction will be taken from the stack and it will be popped. If the conditional test fails, the next microinstruction address will come from the internal microprogram counter. This is depicted in the flow diagram of Figure 1. It is important to remember that every subroutine call must somewhere be followed by a return from subroutine call in order to have an equal number of pushes and pops on the stack.

INSTRUCTION 11 –
CONDITIONAL JUMP PIPELINE AND POP (CJPP)

The Conditional Jump Pipeline and Pop instruction is a technique for exiting a loop from within the middle of the loop. This is depicted fully in the flow diagram for the IDT49C410s, as shown in Figure 1. The conditional test input for this instruction results in a branch being taken if the test is passed. The address selected will be that on the D inputs to the microprogram sequencer and, since the loop is being terminated, the stack will popped. Should the test be failed on the conditional test inputs, the microprogram will simply continue to the next address as taken from the microprogram counter. The stack will not be affected if the conditional test input is failed.

INSTRUCTION 12 –
LOAD COUNTER AND CONTINUE (LDCT)

The Load Counter and Continue instruction is used to place a value on the D inputs in the register/counter and continue to the next microinstruction.

INSTRUCTION 13 –
TEST END OF LOOP (LOOP)

The Test End of Loop instruction is used as a last instruction in a loop associated with the stack. During this instruction, if the conditional test input is failed, the loop branch address will be that on the stack. Since we may go around the loop a number of times, the stack is not popped. If the conditional test input is passed, then the loop is terminated and the stack is popped. Notice that the loop instruction requires a PUSH to be performed at the instruction immediately prior to the loop return address. This is necessary so as to have the correct address on the stack before the loop operation. It is for this reason that the stack pointer always points to the last thing written on the stack.

INSTRUCTION 14 –
CONTINUE (CONT)

The Continue instruction is a simple instruction whereby the address for the microinstruction is taken from the microprogram counter. This instruction simply causes sequential program flow to the next microinstruction in microcode memory.

INSTRUCTION 15 –
THREE WAY BRANCH (TWB)

The Three Way Branch instruction is used for looping while waiting for a conditional event to come true. If the event does not come true after some number of microinstructions, a branch is taken to another microprogram sequence. This is depicted in Figure 1 showing the IDT49C410 flow diagrams and is also described in full detail in the IDT49C410's instruction operational summary. Operation of the instruction is such that any time the external conditional test input is passed, the next microinstruction will be that associated with the program counter and the loop will be left. The stack is also popped. Thus, the external test input overrides the other possibilities. Should the external test input not be true, the rest of the operation is controlled by the internal counter. If the counter

is not equal to 0, the loop is taken by selecting the address on the top of the stack as the address out of the Y outputs of the IDT49C410. In addition, the counter is decremented. Should the external conditional test input be failed and the counter also have counted to 0, this instruction "times out". The result is that the stack is popped and a branch is taken to the address presented to the D inputs of the IDT49C410 microprogram sequencer. This address is usually provided by the external pipeline register.

CONDITIONAL TEST

Throughout this discussion we have talked about microcode passing the conditional test. There are actually two inputs associated with the conditional test input. These include the $\overline{CCEN}$ and the $\overline{CC}$ inputs. The $\overline{CCEN}$ input is a condition code enable. Whenever the $\overline{CCEN}$ input is HIGH, the $\overline{CC}$ input is ignored and the device operates as though the $\overline{CC}$ input were true (LOW). Thus, a fail of the external test condition can be defined as $\overline{CCEN}$ equals LOW and $\overline{CC}$ equals HIGH. A pass condition is defined as a $\overline{CCEN}$ equal to HIGH or a $\overline{CC}$ equal to LOW. It is important to recognize the full function of the condition code enable and the condition code inputs in order to understand when the test is passed or failed.

ABSOLUTE MAXIMUM RATINGS(1)

Symbol	Rating	Com'l.	Mil.	Unit
V_{CC}	Power Supply Voltage	–0.5 to +7.0	–0.5 to +7.0	V
V_{TERM}	Terminal Voltage with Respect to Ground	–0.5 to V_{CC} + 0.5	–0.5 to V_{CC} + 0.5	V
T_A	Operating Temperature	0 to +70	–55 to +125	°C
T_{BIAS}	Temperature Under Bias	–55 to +125	–65 to +135	°C
T_{STG}	Storage Temperature	–55 to +125	–65 to +150	°C
P_T	Power Dissipation	1.0	1.0	W
I_{OUT}	DC Output Current	30	30	mA

NOTE: 2551 tbl 03

1. Stresses greater than those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

CAPACITANCE (T_A = +25°C, f = 1.0MHz)

Symbol	Parameter(1)	Conditions	Typ.	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V	5	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V	7	pF

NOTE: 2551 tbl 04

1. This parameter is sampled and not 100% tested.

DC ELECTRICAL CHARACTERISTICS

Commercial: T_A = 0°C to + 70°C, V_{CC} = 5.0V ± 5%; Military: T_A = – 55°C to + 125°C, V_{CC} = 5.0V ± 10%

Symbol	Parameter	Test Conditions (1)		Min.	Typ. (2)	Max.	Unit
V_{IH}	Input HIGH Level	Guaranteed Logic HIGH Level (4)		2.0	–	–	V
V_{IL}	Input LOW Level	Guaranteed Logic LOW Level (4)		–	–	0.8	V
I_{IH}	Input HIGH Current	V_{CC} = Max., V_{IN} = V_{CC}		–	0.1	5	µA
I_{IL}	Input LOW Current	V_{CC} = Max., V_{IN} = GND		–	– 0.1	– 5	µA
V_{OH}	Output HIGH Voltage	V_{CC} = Min.	I_{OH} = –12 mA MIL	2.4	4.3	–	V
		V_{IN} = V_{IH} or V_{IL}	I_{OH} = –15 mA COM'L.	2.4	4.3	–	
V_{OL}	Output LOW Voltage	V_{CC} = Min.	I_{OL} = 20 mA MIL	–	0.3	0.5	V
		V_{IN} = V_{IH} or V_{IL}	I_{OL} = 24 mA COM'L	–	0.3	0.5	
I_{OZ}	Off State (High Impedance) Output Current	V_{CC} = Max.	V_O = 0V	–	– 0.1	– 10	µA
			V_O = V_{CC} (Max.)	–	0.1	10	
I_{OS}	Output Short Circuit Current	V_{CC} = Max., V_{OUT} = 0V (3)		– 30	–	–	mA

NOTES: 2551 tbl 05

1. For conditions shown as max. or min. use appropriate value specified under Electrical Characteristics.
2. Typical values are at V_{CC} = 5.0V, +25°C ambient and maximum loading.
3. Not more than one output should be shorted at one time. Duration of the circuit test should not exceed one second.
4. These input levels should only be static tested in a noise-free environment.

DC ELECTRICAL CHARACTERISTICS (Cont'd.)

Commercial: T_A = 0°C to + 70°C, V_{CC} = 5.0V ± 5%; Military: T_A = – 55°C to + 125°C, V_{CC} = 5.0V ± 10%

Symbol	Parameter	Test Conditions (1)		Min.	Typ. (2)	Max.	Unit
I_{CCQH}	Quiescent Power Supply Current CP = H (CMOS Inputs)	V_{CC} = Max. V_{IN} = V_{CC} or GND f_{CP} = 0, CP = H		–	35	50	mA
I_{CCQL}	Quiescent Power Supply Current CP = L (CMOS Inputs)	V_{CC} = Max. V_{IN} = V_{CC} or GND f_{CP} = 0, CP = L		–	35	50	mA
I_{CCT}	Quiescent Input Power Supply Current (per Input @ TTL High) (5)	V_{CC} = Max., V_{IH} = 3.4V, f_{CP} = 0		–	0.3	0.5	mA/ Input
I_{CCD}	Dynamic Power Supply Current	V_{CC} = Max. V_{IN} = V_{CC} or GND Outputs Open, $\overline{OE}$ = L	MIL COM'L	– –	1.0 1.0	3.0 1.5	mA/ MHz
I_{CC}	Total Power Supply Current (6)	V_{CC} = Max., f_{CP} = 10MHz Outputs Open, $\overline{OE}$ = L CP = 50 % Duty cycle $V_{HC} \leq V_{IH}$, $V_{IL} \leq V_{LC}$	MIL COM'L	– –	45 45	80 65	mA
		V_{CC} = Max., f_{CP} = 10MHz Outputs Open, $\overline{OE}$ = L CP = 50 % Duty cycle V_{IH} = 3.4V, V_{IL} = 0.4V	MIL COM'L	– –	50 50	90 75	

2551 tbl 06

NOTES:

5. I_{CCQT} is derived by measuring the total current with all the inputs tied together at 3.4V, subtracting out I_{CCQH}, then dividing by the total number of inputs.
6. Total Supply Current is the sum of the Quiescent current and the Dynamic current (at either CMOS or TTL input levels). For all conditions, the Total Supply Current can be calculated by using the following equation:

$I_{CC} = I_{CCQH} (CD_H) + I_{CCQL} (1 - CD_H) + I_{CCT} (N_T \times D_H) + I_{CCD} (f_{CP})$

CD_H = Clock duty cycle high period
D_H = Data duty cycle TTL high period (V_{IN} = 3.4V)
N_T = Number of dynamic inputs driven at TTL levels
f_{CP} = Clock Input frequency

CMOS TESTING CONSIDERATIONS

There are certain testing considerations which must be taken into account when testing high-speed CMOS devices in an automatic environment. These are:

1) Proper decoupling at the test head is necessary. Placement of the capacitor set and the value of capacitors used is critical in reducing the potential erroneous failures resulting from large Vcc current changes. Capacitor lead length must be short and as close to the DUT power pins as possible.
2) All input pins should be connected to a voltage potential during testing. If left floating, the device may begin to oscillate causing improper device operation and possible latchup.
3) Definition of input levels is very important. Since many inputs may change coincidentally, significant noise at the device pins may cause the V_{IL} and V_{IH} levels not to be met until the noise has settled. To allow for this testing/board induced noise, IDT recommends using $V_{IL} \leq 0V$ and $V_{IH} \geq 3V$ for AC tests.
4) Device grounding is extremely important for proper device testing. The use of multi-layer performance boards with radial decoupling between power and ground planes is required. The ground plane must be sustained from the performance board to the DUT interface board. All unused interconnect pins must be properly connected to the ground pin. Heavy gauge stranded wire should be used for power wiring and twisted pairs are recommended to minimize inductance.

10

IDT49C410A
AC ELECTRICAL CHARACTERISTICS
I. SET-UP AND HOLD TIMES

Inputs	t (S)		t (H)		Unit
	Com'l.	Mil.	Com'l.	Mil.	
DI → R	6	7	0	0	ns
DI → PC	13	15	0	0	ns
I 0–3	23	25	0	0	ns
$\overline{CC}$	15	18	0	0	ns
$\overline{CCEN}$	15	18	0	0	ns
CI	6	7	0	0	ns
$\overline{RLD}$	11	12	0	0	ns

2551 tbl 07

II. COMBINATIONAL DELAYS

Inputs	Y		$\overline{PL}$, VECT, MAP		$\overline{FULL}$		Unit
	Com'l.	Mil.	Com'l.	Mil.	Com'l.	Mil.	
D0–11	12	15	–	–	–	–	ns
I 0–3	20	25	13	15	–	–	ns
$\overline{CC}$	16	20	–	–	–	–	ns
$\overline{CCEN}$	16	20	–	–	–	–	ns
CP	28	33	–	–	22	25	ns
$\overline{OE}$ (1)	10/10	13/13	–	–	–	–	ns

2551 tbl 08

NOTE:
1. Enable/Disable. Disable times measure to 0.5V change on output voltage level with C_L = 5pF. Tested at C_L = 50pF, correlated to 5pF.

III. CLOCK REQUIREMENTS

	Com'l.	Mil.	Unit
Minimum Clock LOW Time	18	20	ns
Minimum Clock HIGH Time	17	20	ns
Minimum Clock Period	35	40	ns

2551 tbl 09

IDT49C410
AC ELECTRICAL CHARACTERISTICS
I. SET-UP AND HOLD TIMES

Inputs	t (S)		t (H)		Unit
	Com'l.	Mil.	Com'l.	Mil.	
DI → R	16	16	0	0	ns
DI → PC	30	30	0	0	ns
I 0–3	35	38	0	0	ns
$\overline{CC}$	24	35	0	0	ns
$\overline{CCEN}$	24	35	0	0	ns
CI	18	18	0	0	ns
$\overline{RLD}$	19	20	0	0	ns

2551 tbl 10

II. COMBINATIONAL DELAYS

Inputs	Y		$\overline{PL}$, VECT, MAP		$\overline{FULL}$		Unit
	Com'l.	Mil.	Com'l.	Mil.	Com'l.	Mil.	
D0–11	20	25	–	–	–	–	ns
I 0–3	35	40	30	35	–	–	ns
$\overline{CC}$	30	36	–	–	–	–	ns
$\overline{CCEN}$	30	36	–	–	–	–	ns
CP	40	46	–	–	31	35	ns
$\overline{OE}$ (1)	25/27	25/30	–	–	–	–	ns

2551 tbl 11

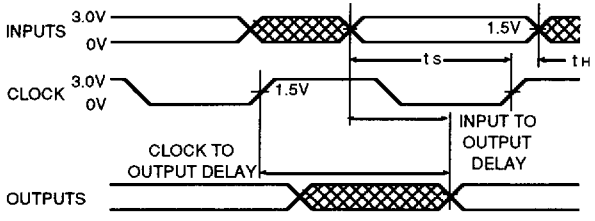
NOTE:
1. Enable/Disable. Disable times measure to 0.5V change on output voltage level with C_L = 5pF. Tested at C_L = 50pF, correlated to 5pF.

III. CLOCK REQUIREMENTS

	Com'l.	Mil.	Unit
Minimum Clock LOW Time	20	25	ns
Minimum Clock HIGH Time	20	25	ns
Minimum Clock Period	50	51	ns

2551 tbl 12

SWITCHING WAVEFORMS



2551 drw 06

IDT49C410 INPUT/OUTPUT INTERFACE CIRCUITRY

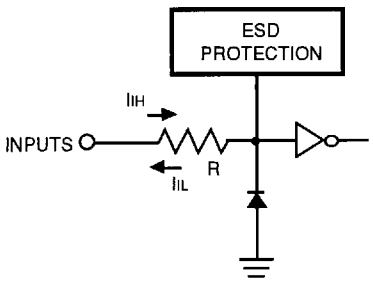


Figure 2. Input Structure

2551 drw 07

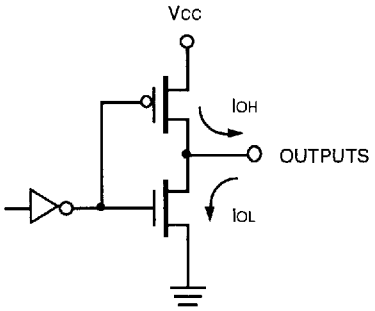
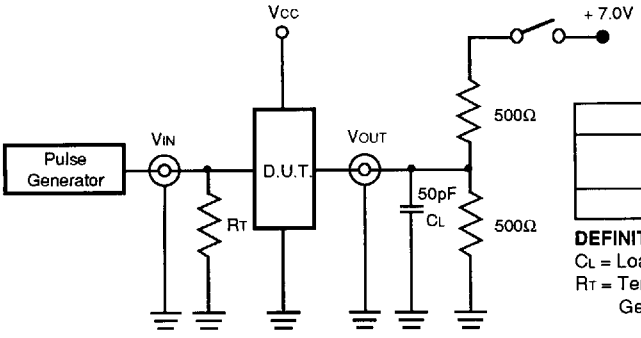


Figure 3. Output Structure

2551 drw 08

TEST LOAD CIRCUIT



Test	Switch
Disable Low Enable Low	Closed
All other Tests	Open

2551 tbl 13

DEFINITIONS:

C_L = Load capacitance: includes jig and probe capacitance

R_T = Termination resistance: should be equal to Z_{OUT} of the Pulse Generator

Figure 4. Switching Test Circuits

2551 drw 09

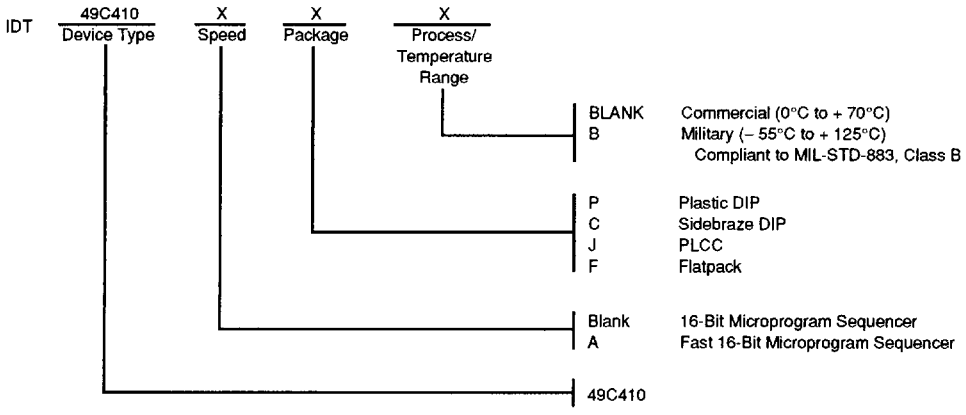
AC TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise/Fall Times	1V/ns
Input Timing Reference Levels	1.5V
Output Reference Levels	1.5V
Output Load	See Figure 3

2551 tbl 14

10

ORDERING INFORMATION



2551 drw 10