

ENGINEERING APPROACH TO DIGITAL DESIGN



Prentice/Hall International editions

WILLIAM I. FLETCHER

ROYAL COUNTY OF BERKSHIRE



MAIDEN ERLEGH SCHOOL

ACADEMIC PRIZE

1983 - 84

Presented to ROBERT ARCHER

For OUTSTANDING ACHIEVEMENT AT 'A' LEVEL

AN ENGINE
TO DIGITAL

*AN ENGINEERING APPROACH
TO DIGITAL DESIGN*

Library of Congress Cataloging in Publication Data

FLETCHER, WILLIAM I

An engineering approach to digital design.

Includes bibliographies and index.

1.-Digital electronics. 2.-Logic design.

3.-Machine theory. I.-Title.

TK7868.D5F5 621.3815 78-27177

ISBN 0-13-277807-6

This edition may be sold only in those countries
to which it is consigned by Prentice-Hall International.
It is not to be re-exported and it is not for sale
in the U.S.A., Mexico or Canada.

© 1980 by Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632

All rights reserved. No part of this book
may be reproduced in any form or
by any means without permission in writing
from the publisher.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Editorial/production supervision
and interior design by *Virginia Huebner*
Page layout by *Gail Cocker*
Manufacturing buyer: *Gordon Osbourne*

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*
PRENTICE-HALL, INC., *Englewood Cliffs, New Jersey*

CONTENTS

Preface

xiii

Chapter 1 Introductory Digital Design Concepts

1

1-1	Introduction	1
1-2	WHY and WHEN Digital	2
1-3	Digital and its Symbolology	8
1-4	Digital and the Real World	10
1-5	Digital-to-Analog/Analog-to-Digital Converters	12
1-6	Number Systems	19
1-7	Juxtapositional Numbers	19
1-8	Polynomial Notation	23
1-9	Base Conversion Methods	23
1-10	Number Conversion	24
1-11	Complements of Numbers	32
1-12	Codes	35
1-13	Binary Codes	36
1-14	Binary Coded Decimal Codes	38
1-15	BCD Code Properties	39
1-16	Unit Distance Codes	40
1-17	Alphameric Codes	44
1-18	Error Codes	46
1-19	Other Error Detecting and Correcting Codes	48
1-20	Once Over Lightly	48
1-21	Summary	52

vii

Chapter 2 Digital Design Fundamentals**57**

2-1	Introductory Comments	57
2-2	Introductory Concepts of Digital Design	58
2-3	The Truth-Table	62
2-4	A Word About "Gates" and "Functions"	65
2-5	The Logic Operators	67
2-6	Hardware Aspects Related to ASSERTED and NOT-ASSERTED Conditions	73
2-7	Using Mechanical Switches for Signal Sources	76
2-8	The Concept of the INVERTER	77
2-9	General Implementation Procedures	80
2-10	Axiomatic Systems and Boolean Algebra	83
2-11	Other Gates and Their Functions	91
2-12	NAND and NOR Gates	92
2-13	The AND, OR, and INVERTER Functions of the NOR Gate	94
2-14	Justification of Distinctive-shaped Symbol Set	98
2-15	EXOR and EQUIVALENCE Gates	98
2-16	MINTERM and MAXTERM Realizations	100
2-17	MAXTERM Representation	104
2-18	Once Over Lightly and Combination Analysis	108
2-19	Analysis of Combinational Circuits	118
2-20	Summary	121

Chapter 3 Minimization and Design of Combinational Circuits**129**

3-1	Introduction	129
3-2	Minimization with Theorems	130
3-3	The Karnaugh Map	134
3-4	Plotting a Karnaugh Map	135
3-5	Labeling a K-Map	137
3-6	Reading the K-Map	140
3-7	Five and Six Variable Maps	144
3-8	Prime and Essential Implicants	146
3-9	Don't Care Map Entries	148
3-10	Map Reduction Resulting in Product-of-Sum Expressions	153
3-11	Using the Map for Simplifying Partially Simplified Expressions	154
3-12	Multi-Output Minimization	155
3-13	Variable-Entered Mapping	157
3-14	VEM Plotting Theory	158
3-15	VEM Reading Theory	162
3-16	Minimization and Combinational Design	167
3-17	Minimizing Functions of More than Six Variables	174
3-18	Introduction to the "BOOZER" Program	176
3-19	Using BOOZER	183
3-20	Summary	185

Chapter 4**Chapter 5**

4-1	Introduction	198
4-2	Examples of Useful Digital Circuits	200
4-3	Arithmetic Circuits	200
4-4	Comparators	207
4-5	Multiplexers	210
4-6	Code Converters	226
4-7	EXOR and AND-OR-INVERT Gates	244
4-8	Hints for Reading EXOR and NEXOR Expressions Directly from the Map	247
4-9	AND-OR-INVERT Gates (A-O-I)	250
4-10	Wired Logic	251
4-11	Practical Aspects of Wired Logic and Bus-Oriented Structures	254
4-12	Tri-State Bus Systems	256
4-13	Practical Aspects Related to Combinational Logic Design	259
4-14	FAN-IN FAN-OUT	259
4-15	Propagation Delay	261
4-16	Summary	265

Chapter 5 Sequential Machine Fundamentals

5-1	Introduction	275
5-2	The Need for Sequential Circuits	276
5-3	Basic Architectural Distinctions between Combinational and Sequential Circuits	280
5-4	The Concept of Memory	281
5-5	The Binary Cell	283
5-6	The CELL and the Bouncing Switch	287
5-7	Fundamental Differences between Sequential Machines	290
5-8	Fundamentals of Sequential Machine Operation	290
5-9	Using Variations of the General Model to Classify Sequential Machines	293
5-10	The Flip-Flop	295
5-11	CLOCK and OSCILLATORS	297
5-12	Types of Traditional Clocked Flip-Flops	302
5-13	The SET/RESET Flip-Flop	302
5-14	The "D" Latch Flip-Flop	304
5-15	The "CLOCKED T" Flip-Flop	305
5-16	The "CLOCKED JK" Flip-Flop	306
5-17	The Design of a Clocked Flip-Flop	308
5-18	Flip-Flop Conversion from One Type to Another	311
5-19	Practical Clocking Aspects Concerning Flip-Flops	314
5-20	Timing and Triggering Considerations	323
5-21	Clock Skew	324
5-22	Summary	328

Chapter 6 Traditional Approaches to Sequential Analysis and Design 335

6-1	Introduction	335
6-2	The State Diagram	336
6-3	Analysis of Synchronous Sequential Circuits	339
6-4	A Synchronous Analysis Process	340
6-5	Approaches to the Design of Synchronous Sequential Finite State Machines	353
6-6	Design Steps for Traditional Synchronous Sequential Circuits	353
6-7	State Reduction	364
6-8	Minimizing the Next State Decoder	368
6-9	Design Steps Leading to Next State Decoders	369
6-10	Trying a JK or T Flip-Flop	375
6-11	Output Decoder Design	381
6-12	Counters	386
6-13	Design of Single-Mode Counters	391
6-14	Multi-Mode Counters	392
6-15	Design of Specialized Multi-Mode Counters	394
6-16	Ripple Counters	398
6-17	Ring Counters	402
6-18	Applications of Ring Counters	406
6-19	Shift Register	407
6-20	Shift Register Sequences	415
6-21	Ring Counters Using Shift Register	416
6-22	Shift Registers and Memory	418
6-23	Final Notes	424
6-24	Summary	424

Chapter 7 Introduction to Multi-Input System Controller Design 440

7-1	Introduction	440
7-2	System Controllers	441
7-3	Controller Design Phases and System Documentation	444
7-4	Defining the Purpose and Role of the System	445
7-5	Defining the Basic Operational Characteristic of the System, the Controlling Systems, and the Controlled System	447
7-6	Timing and Frequency Considerations	451
7-7	Functional Partition and Detailed Flow Diagram Development	452
7-8	System Controller State Specification (MDS Diagram Generation)	457
7-9	Synchronizing Two Systems and Choosing Controller Architecture	465
7-10	The State Assignment	475
7-11	The Next State Decoder	485
7-12	Next State Decoder Maps	501
7-13	The Output Decoder	503
7-14	Clock Frequency Determination	506
7-15	Power Supply Requirements	506
7-16	Control and Display	507
7-17	Concepts Related to the Use of Conditional Outputs	510
7-18	Summary	514

x Contents

Chapter 8 System

8-1	Intro
8-2	Usin
8-3	Usin
8-4	Syste
8-5	Indir
8-6	Read
8-7	ROM
8-8	Using
8-9	Progr
8-10	Appli
8-11	Syste
8-12	Indir
8-13	Syste
8-14	Sumn

Chapter 9 Introduct

9-1	Introd
9-2	System
9-3	Design
9-4	System
9-5	The Co
9-6	Some l
9-7	Genera
9-8	Notes
9-9	Progra
9-10	Progra
9-11	MYCA
9-12	The 82
9-13	The A1
9-14	The 72
9-15	Summa

Chapter 10 Asynchron

10-1	Introc
10-2	Why .
10-3	Scope
10-4	Async
10-5	The I
10-6	Cycle:
10-7	Plottin
10-8	Hazar
10-9	Essen

Chapter 8 System Controllers Utilizing Combinational MSI/LSI Circuits 532

8-1	Introduction	532
8-2	Using MSI Decoder in System Controllers	533
8-3	Using MSI Multiplexers in System Controllers	537
8-4	System Controllers with More Complexity	540
8-5	Indirect Addressed Multiplexers Configuration	544
8-6	Read Only Memories (ROM's)	546
8-7	ROM's, PROM's and Applications	548
8-8	Using a PROM for Random Logic	557
8-9	Programmed Logic Arrays (PLA's)	561
8-10	Applications of PLA's and FPLA's	568
8-11	System Controller Designs Centered around ROM's and PROM's	568
8-12	Indirect-Addressed Multiplexer/ROM Configuration	570
8-13	System Controller Designs Centered around an FPLA	575
8-14	Summary	578

Chapter 9 Introduction to Programmable System Controllers 580

9-1	Introduction	580
9-2	System Controllers Centered around an MSI Shift Register	582
9-3	Design and Implementation Practices Using the AM74LS194	587
9-4	System Controllers Centered around an MSI Counter	592
9-5	The Concepts of a Programmable System Controller	600
9-6	Some Basic Desirable Features of a Programmable Controller	600
9-7	General Requirements	600
9-8	Notes Related to the Micro-Instructions	605
9-9	Program Controllers with Fixed Instruction Set	608
9-10	Programmable Sequence Controllers with Sub-Routine Capability	613
9-11	MYCA-I Plus an ADDER and STACK	615
9-12	The 8X02 Control Store Sequencer	619
9-13	The Application of the 8X02 to System Control Functions	626
9-14	The 74S482 and MC2909 Integrated Circuit Sequence Controllers	629
9-15	Summary	644

Chapter 10 Asynchronous Finite State Machines 649

10-1	Introduction	649
10-2	Why Asynchronous Circuits	652
10-3	Scope	653
10-4	Asynchronous Analysis	653
10-5	The Design of Asynchronous Machines	661
10-6	Cycles and Races	668
10-7	Plotting and Reading the Excitation Map	670
10-8	Hazards	674
10-9	Essential Hazards	678

10-10 More Worked Examples	684
10-11 The Map-Entered-Variable Approaches to Asynchronous Design	692
10-12 A Contemporary Approach to Asynchronous Design	700
10-13 Hazards in Circuits Developed by MEV Method	703
10-14 Summary	710

Appendix A Drafting and Documentation Standards Manual	718
---	------------

Appendix B Boozer Program	743
----------------------------------	------------

Index	761
--------------	------------

PREF

DIGITAL D

Sinc
in 1
cept
and
expc
only
price
facil
using
rapic
to re
Certa
will

micro
gram
micro
direc
micro
desig
advan

ating the
The flow
states in
ate, all in
8.3. Note
on the low
nimize the

FLAG=1 and $\overline{\text{CLK}}$, all of which can be traced through the detailed flow diagram.

Now consider one more design problem illustrating the use of the design phases (1-9). This problem requires you to interpret a set of detailed specifications and further illustrate the functionality of the system controller concept. Special thanks to Prof. Richard Ohran of Brigham Young University for suggesting a problem such as this back in 1971.

EXAMPLE 7-5: The Pop Machine Controller

Introduction and specifications. The El-Rip-O Vending Machine Company wishes to update its Model 1909 mechanical vending machine. It is desired that a first generation digital controlled prototype system be developed for test and evaluation; nothing really fancy, just an evaluation prototype. The El-Rip-O Company has entered into an OEM (Original Equipment Manufacturers) agreement with the Futzel-Boopsdink Pipe and Die Company to provide the coin receiver, coin changer, and pop dropping systems for the new El-Rip-O Model 1971 system when it is fully developed.

DE)

INVERT
K]

Preliminary specifications. The initial digital control system should be developed such that it will direct the control of the coin receiver, coin changer, and pop drop mechanics and provide the El-Rip-O Company with a system capable of automatically dispensing soda pop at 30¢ per can and making the proper change retrieval for the following coin sequences of nickels, dimes, quarters, and half-dollars.

Prescribed operation. Upon the insertion of each coin, the controller is to record the coin value and issue the proper change, if required, in nickels only, then drop a can of soda pop. At this time a solenoid is to be activated that lets the coins collected in the coin sequence drop into a common collector box. This system is to have a coin release feature for the manual release of jammed coins and coins collected in "short" sequences, and the coin release feature is also intended to reset the system to an initial condition.

Coin receiver specification. The Futzel-Boopsdink Pipe and Die Model CR-1971 coin receiver is described by the following features:

- mentation for
o general but
ted here is a
controller and
important that
flow diagram.
ion ends with
itions are de-
cision paths
to all other
nes the condi-
for SET IN-
T INVERT is
SET and BIT
- (1) single slot coin entry;
 - (2) electronic coin detection;
 - (3) guaranteed coin detection for U.S. half dollars, quarters, dimes, and nickels;
 - (4) automatic nonvalid and bent coin rejection;
 - (5) all control inputs and outputs are standard TTL compatible;
 - (6) coin-catching mechanism and manual coin release feature;
 - (7) a special mechanical mechanism that prevents coin over-run.

Electrical specifications. See Figure 7-32.

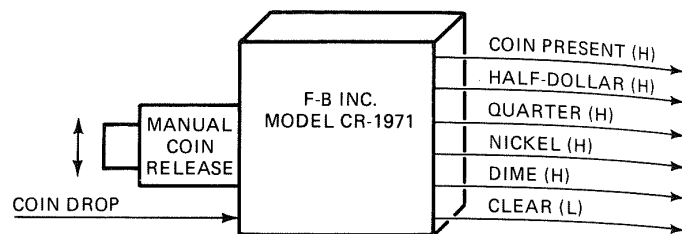


Fig. 7-32. Block diagram of CR-1971.

Signal description.

COIN DROP—an input that commands the CR-1971 to drop collected coins into the common collection box.

CLEAR—an output that will go to a 5 volt ± 0.5 level for the duration of the depression of the coin release.

COIN PRESENT—an output signifying that a coin is present in the coin receiver and denomination has been determined.

HALF DOLLAR, QUARTER, DIME and NICKEL—outputs signifying the denomination of coin present.

The timing relationship between **COIN PRESENT**, **HALF-DOLLAR**, **QUARTER**, **DIME**, and **NICKEL** is described by the timing diagram shown in Figure 7-33.

Coin changer. The Futzel-Boopsdink Pipe and Die Model CC-1971 coin changer (see Figure 7-34), developed by the Stuck-Muzzey subsidiary, features:

- (1) fast, reliable, electro-mechanical nickel-ejecting system (100 msec per nickel maximum);
- (2) special **READY** status output which indicates when another coin ejection sequence can be started;
- (3) automatic load of 50 nickels reserve.

EJECT NICKEL—TTL compatible input, pulse triggered with minimum width of 10 msec restriction.

CHANGER READY—an output status line indicating the changer will respond to **EJECT NICKEL**.

Timing relations. See Figure 7-35.

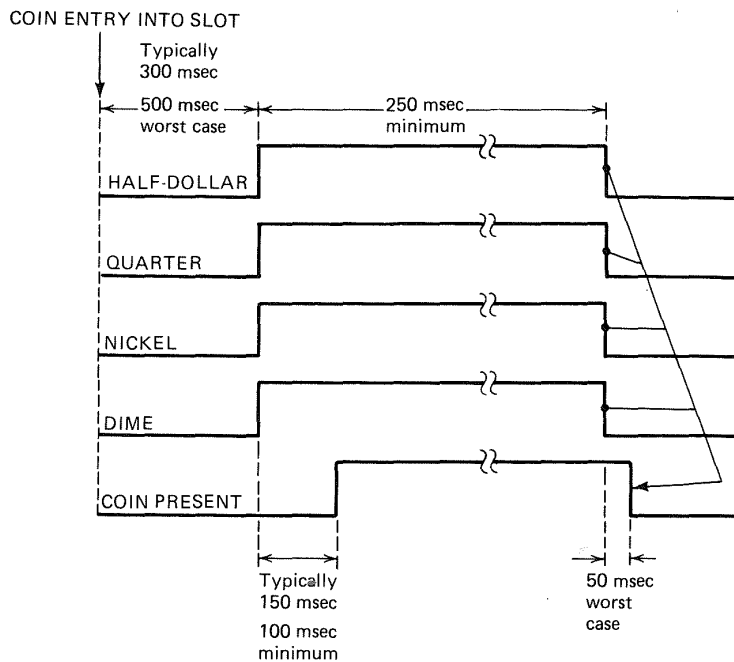


Fig. 7-33. Timing detail of the CR-1971.

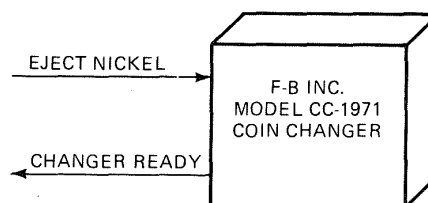


Fig. 7-34. The block diagram of the CC-1971.

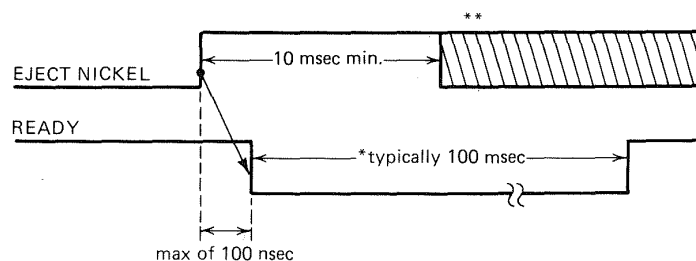


Fig. 7-35. Timing detail of the CC-1971.

Pop-drop mechanism The Futzel-Boopsdink Model PD-1971 pop-drop mechanism developed by Bend-A-Can Division (see Figure 7-36) is a fast solenoid operated device featuring:

- (1) single TTL compatible drop command input;
- (2) pulse operated with 10 msec minimum pulse width;
- (3) status READY line

Timing and operation is identical to that of the coin changer.

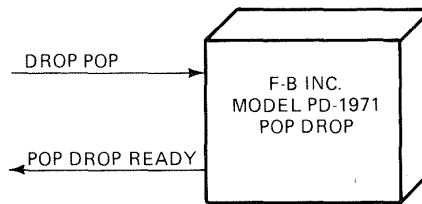


Fig. 7-36. The block diagram of the PD-1971. Note that POP DROP READY will not return to the READY status if the mechanism is jammed.

7-11.3 - SYSTEM DEVELOPMENT

First let it be noted that the system to be designed has certain frailties related to its cheating the customer as well as the customer cheating it. However, since this development is only an evaluation prototype, these frailties can be temporarily overlooked. The specifications for a foolproof system are given as a design exercise and left for you to work out the fine details.

Phase I. Because of your intimate knowledge of vending machines derived from the everyday encounters with these beasts (see Figure 7-37), along with the specifications given, Phase I is pretty well covered.



Fig. 7-37. Man and the vending machine.

Phase II. From the knowledge accounted for in Phase I, the bare-bones block diagram as shown in Figure 7-38 can be readily developed. This bare-bones block diagram is supported by the first-cut flow diagram shown in Figure 7-39. Note that this flow diagram clearly defines the proposed big picture sequential behavior of the controller operation. Now keep in mind that this proposed operation is but one of several possible alternatives. However, it was selected on the basis of the following criteria:

- (1) It provides a practical and effective system.
- (2) The author feels that it nicely demonstrates the logical sequence of events.
- (3) It also demonstrates how properly designed "hard wired" systems can have a reasonable degree of flexibility.

Though these criteria are somewhat biased, the concept of making a selection from several possible alternatives should not be considered as such. It is very important that you derive several alternatives and then select one of these based on some criteria. Now these criteria can vary depending on the design requirements, but the major cornerstones should be:

- (1) Practicality
- (2) Effectiveness
- (3) Efficiency
- (4) Degree of flexibility

The flexibility facet should be treated in the light of possible future applications and modifications. This is in keeping with the invariable fact that somebody will want something changed at the most inopportune time. However, don't let these anticipations completely override your preliminary preparations; just keep them in mind. For example, though there is no mention of price changes or multi-pop selection requirements in the specifications for the pop machine, these features are included with no real extra effort or cost. They are simply a natural fallout of the proposed system.

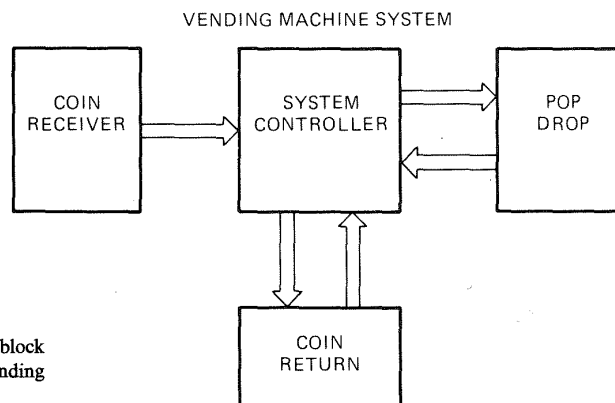


Fig. 7-38. The bare-bones block diagram of the prototype vending machine.

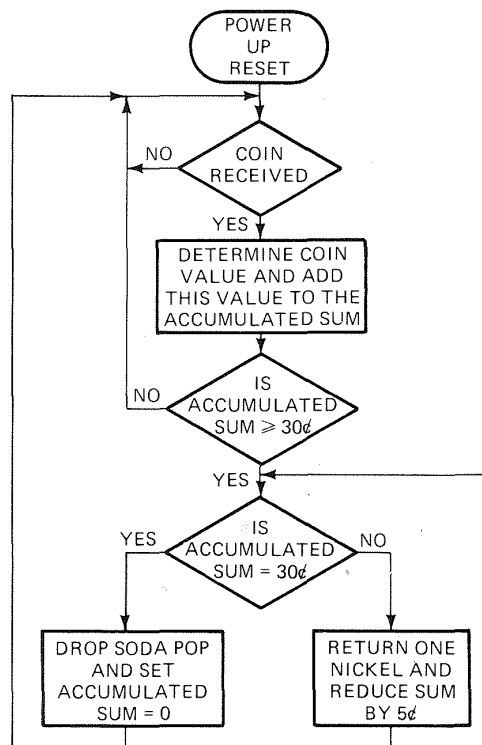


Fig. 7-39. A proposed first-cut flow diagram defining the vending machine's basic operation.

7-11.4 THE DETAILED FLOW DIAGRAM, FUNCTIONAL PARTITION AND MDS DIAGRAM DESCRIPTION

The detailed flow diagram shown in Figure 7-40 defines the exact sequential behavior of the functional partition shown in Figure 7-41. It should be clear that this flow diagram is an extension to the first-cut diagram. Along with this it also defines the timing relations of the control signal to and from the subsystems. The functional partition typifies the system controller in its presiding role. It should be noted that the arithmetic operations called for are carried out by the interconnection of the ADDER, PIPO REGISTER, DOWN COUNTER, and COMPARATOR. The PIPO REGISTER was added to insure the edge-triggered loading of the COUNTER. This register/counter complex is referred to as the ACCUMULA-

TOR (ACC) in this example. Further, it should be noted that a "strappable" compare word allows for easy price changing up to 75¢. The value of each coin is then encoded to the binary value of the number of nickels it represents in accordance with the indicated encoding (w_4, w_3, w_2, w_1) specification table. This encoded value is then added to the present value of the accumulator on the rising-edge of COIN PRESENT. Thus the basic functions called for by the first-cut and detailed flow diagram are brought to light with a hardware description. Note further that the IC package count is up to six packages, all with 16 pins or less.

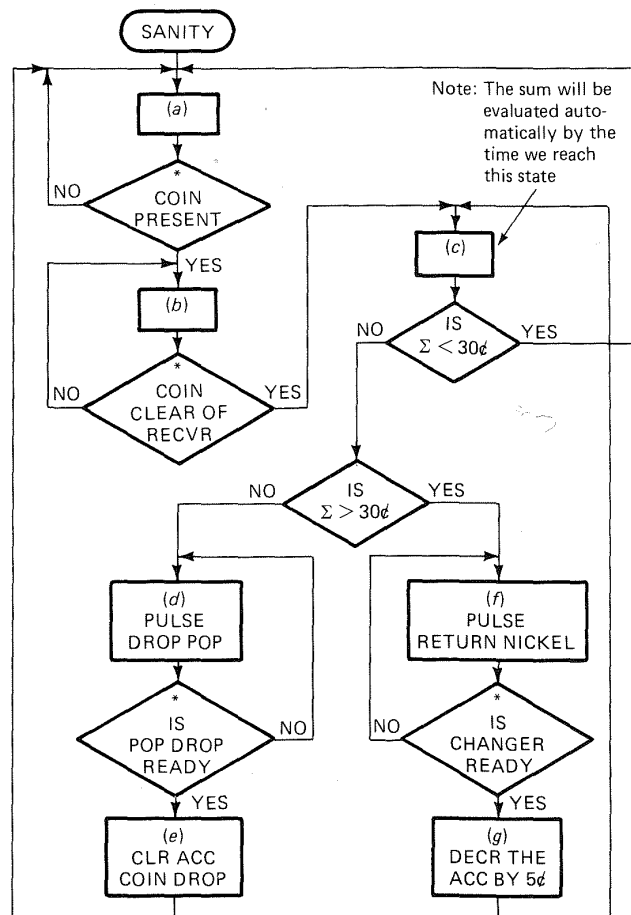


Fig. 7-40. The detailed flow diagram for the prototype pop machine system controller.

act sequential
l be clear that
ith this it also
systems. The
. It should be
: interconnec-
COMPARA-
loading of the
CCUMULA-

w_4 w_3
 CODE = 1 0
 CODE = 0 1
 CODE = 0 0
 CODE = 0 0

The seven-state MDS diagram for the system controller is shown in Figure 7-42. All branching conditions and outputs are clearly specified in close accordance with the specification set forth in the detailed flow diagram. The branching decisions based on asynchronous inputs are properly identified and are defined by the GO, NO-GO decision construct discussed in Section 7-10.1

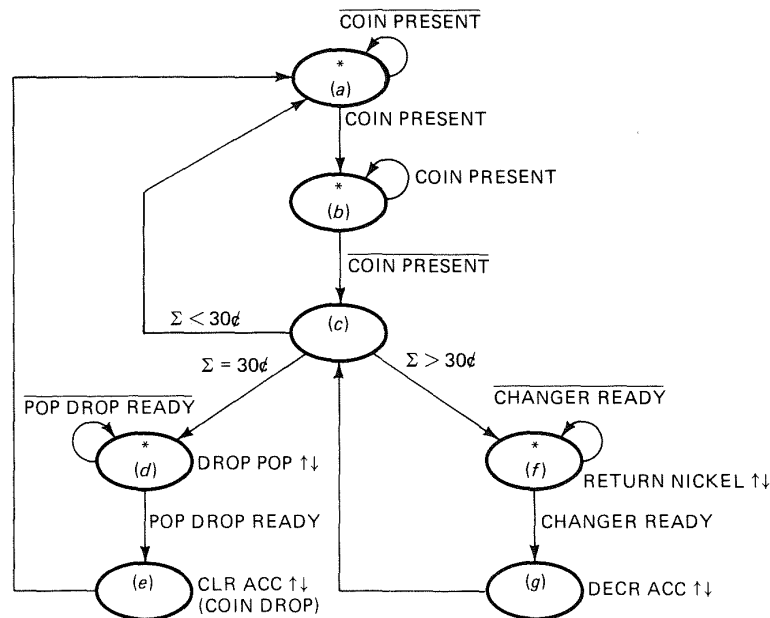


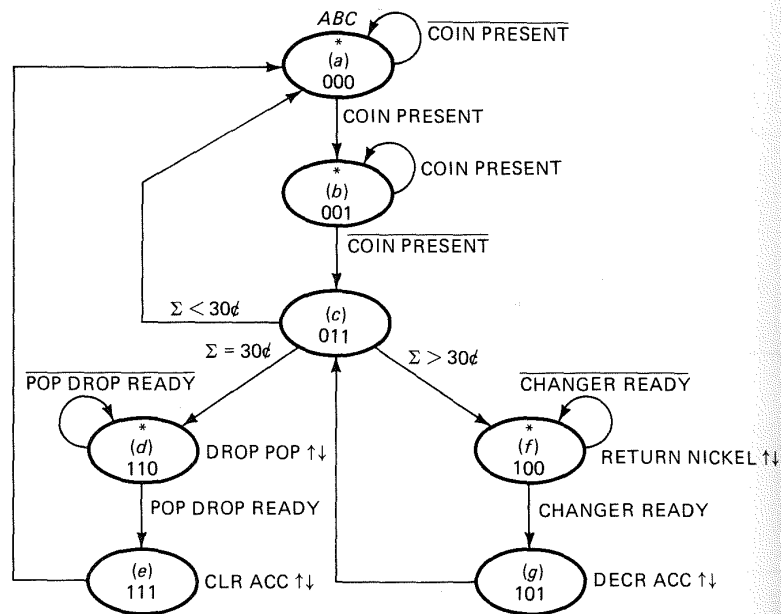
Fig. 7-42. The MDS diagram for the prototype pop machine system controller. Note that it is assumed CLR ACC ↑↓ causes the coins to drop.

7-11.5 THE HARDWARE IMPLEMENTATION OF THE NEXT STATE AND OUTPUT DECODER

Up to this point the major effort has been directed toward the system's three D's: Definition, Description, and Documentation. Now some of the practical aspects of an actual hardware development process are presented. The first step in this development process is to make a state assignment in accordance with the rules set forth earlier, paying strict attention to asynchronous branching (see Figure 7-43). A state map (regular K-map) is used to aid the state assignment process. Here the rules related to asynchronous branching variables are given the highest priority.

This priority calls for the following state assignment constraints:

- a* adjacent to *b*
- b* adjacent to *c*
- d* adjacent to *e*
- f* adjacent to *g*



C \ AB	00 01 11 10			
	0	2	6	4
0	a	∅	d	f
1	b	c	e	g

State Map

Fig. 7-43. The system controller MDS diagram for the prototype pop machine complete with a state assignment and state map.

The rest of the states are assigned with some reference to the simple rules set forth in Chapter 6 and the list in Table 7-1. However, no exhaustive attempt is made to optimize the state assignment.

7-12 NEXT

Once a commit with w/ Flip-Flop about i ENTEI PRESE NEXT does re inspect diagram

the N the D proce attent

nts:

TABLE 7-1 State assignment listing

Previous States	State in Question	Next State	Assignment Suggestions
a, c, e	a*	b	a ADJ b*; a, c, e ADJ
a, b*	b*	b, c	b ADJ c*; a ADJ b*
b	c	a, d, f	a, d, f ADJ; b ADJ c*
c, d	d*	e	d ADJ e*; c ADJ d
d	e	a	NR; d ADJ e*
c, f	f*	g	f ADJ g*; c ADJ f
f	g	c	NR; f ADJ g*

*→a must assignment; NR→NO Requirement

7-12 NEXT STATE DECODER MAPS

Once a state assignment is made, the next design step determines the first real commitment to any specific type of hardware. This example demonstrates the ease with which small sequential machines can be implemented at the SSI gate and Flip-Flop level, knowing well that other alternatives do exist. This ease is brought about in part by taking advantage of the inherent application of the VARIABLE-ENTERED MAP. Using VEM's bypasses the tedious detail of filling out a PRESENT/NEXT STATE TABLE by going *directly from the MDS diagram to the NEXT STATE maps*. The plotting of the VEM's is a simple process. However, it does require a *properly documented MDS diagram* and some ability to recognize, by inspection, the **CONDITIONAL SET OPERATIONS** called for in this MDS diagram. This process is described in the following paragraph.

Next State Map Plotting Process

Using a MDS diagram with a state assignment, select a state, then examine the branching mnemonics to *all* NEXT STATES of the selected state; and enter into the STATE CELL of the NEXT STATE MAP the Boolean expression that defines all of the SET CONDITIONS (0→1, 1→1) that are required for the chosen state variable in that state. Thus these **CONDITIONAL SET EXPRESSIONS** become **MAP ENTERED VARIABLES** in the NEXT STATE MAPS.

This process is repeated for each STATE VARIABLE of each STATE until the NEXT STATE maps are completed. Remember, the NEXT STATE maps are the D-maps from which JK and T maps can be readily obtained. The described processes are clearly demonstrated in Figures 7-44 and 7-45. Pay particular attention to the entry in state cell c in the D_A map. Here you see:

$$(= + >)$$

which indicates that the A Flip-Flop is set by the next clock edge if the machine is in: STATE c AND ($=$) OR ($>$) are ASSERTED. The rest of the MEV ENTRIES are obvious. However, it is recommended that a map be generated for each Flip-Flop and these maps be plotted in parallel, one state at a time, using the state map as a reference for the state cell location.

From Figures 7-44 and 7-45 notice that the JK does yield a circuit implementation requiring fewer gates (11 gates or four IC packages). Thus the new grand total for the system equals nine IC packages, some with spare gates, which really isn't too bad considering the capabilities of the overall system. However, the design for the OUTPUT DECODER must be completed.

C	AB			
	00	01	11	10
0	0 ⁰	$\emptyset$ ²	1 ⁶	1 ⁴
1	0 ¹	$=+>$ ³	0 ⁷	0 ⁵

$$D_A = \bar{A}B(=+>) + A\bar{C}$$

C	AB			
	00	01	11	10
0	0 ⁰	$\emptyset$ ²	1 ⁶	0 ⁴
1	$\bar{C}P$ ¹	$=$ ³	0 ⁷	1 ⁵

$$D_B = \bar{A}B(=) + \bar{B}C(\bar{C}P) + B\bar{C} + A\bar{B}C$$

C	AB			
	00	01	11	10
0	CP ⁰	$\emptyset$ ²	PDR ⁶	CR ⁴
1	1 ¹	0 ³	0 ⁷	1 ⁵

$$D_C = \bar{A}\bar{B}(CP) + \bar{B}\bar{C}(PDR) + A\bar{B}(CR) + \bar{B}C$$

Legend: CP = COIN PRESENT
 $=+> = (\Sigma = 30¢) + (\Sigma > 30¢)$
 CR = CHANGER READY
 PDR = POP DROP READY

Fig. 7-44. The NEXT STATE or D maps for the proto-type pop machine system controller.

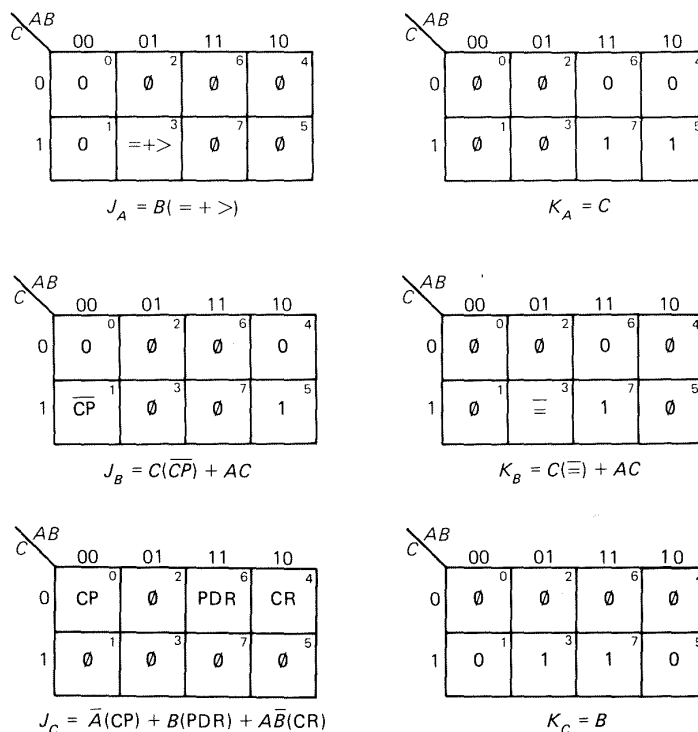
7-13 THE C

The d
design

Howev
system
This i
Theref
output

ie machine is
of the MEV
generated for
me, using the

a circuit im-
Thus the new
gates, which
However, the



Requires 11 gates, or four IC packages with spare gates.

Fig. 7-45. The NEXT STATE DECODER maps for a JK implementation.

7-13 THE OUTPUT DECODER

The design of the OUTPUT DECODER is like any other combination decoder design problem that can be summed up quickly by the following process:

Make up a map for *each* output. Plot the map using MEV for the conditional variables, then simplify and implement.

However, many times maps serve no useful purpose. For example, in the present system controller there is no output that is a function of more than one (1) state. This information can be quickly gleaned from inspecting the MDS diagram. Therefore, in cases such as these, it is recommended that you make up a simple output list. This output list is nothing more than a list of every required output and

the states and conditions related to the generation of these outputs. For example, the MDS diagram in Figure 7-43 would have the following list:

$$\text{RETURN NICKEL} = A\bar{B}\bar{C}$$

$$\text{DECR ACC} = A\bar{B}C$$

$$\text{DROP POP} = ABC$$

$$\text{CLR ACC} = ABC$$

From this expression list observe that none can be further simplified. Therefore, each requires a simple three-input gate (AND function) with the inputs connected to the PRESENT STATE VARIABLES (ABC). However, the ever present glitch problem exists because of our state assignment. The transition from 111→000 introduces the possibility of a glitch in three states. But the real question is which three states. Previous studies indicate that all six of the remaining states are vulnerable when the 111→000 transition is made. Therefore, some sort of glitch prevention must be introduced. In cases with a smaller number of bits changing (two in this case), the possible transitions can be traced in a state map for easy identification of possible glitch problems. However, the 111→000 transition overrides the need for any further glitch studies.

Since there are no outputs which must remain ASSERTED over several states, an OUTPUT HOLDING REGISTER would appear to be an overkill. However, DROP POP and RETURN NICKEL generated in states D and F must be held stable all during these states. Therefore, a holding register is probably called for. Therefore, four three-input NAND gates (74LS10) are chosen for the output decoder and a 74LS175 is chosen as the OUTPUT HOLDING REGISTER. For the completed schematic of the total system controller including the OUTPUT DECODER, see Figure 7-46. From this figure observe that the grand total IC package count is 14, all of which are standard or established 14 or 16 pin IC's (see parts list in Table 7-2).

TABLE 7-2 Parts List

Quantity	Number	Description
1	74LS283	Four-bit adder
1	74LS193	Hex up/down counter
2	74LS175	Quad D PIPO
2	74LS112	Dual edge-triggered JK Flip-Flop
1	74LS85	Four-bit comparator
2	74LS32	Quad of two-input OR
2	74LS10	Triple three-input NAND
1	74LS08	Quad of two-input AND
*1	7404	Hex of inverters
1	74LS00	Quad of two-input NAND
14		

*The standard 7404 was chosen because of the high drive requirement of the clock input on the 74LS112 (four std LS loads per input)

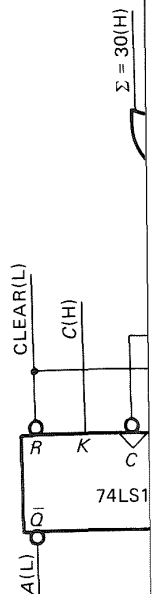


Fig. 7-46. T system.

utputs. For example
st:

er simplified. There
on) with the inputs
However, the ever
The transition from
but the real question
the remaining states
, some sort of glitch
er of bits changing
state map for easy
000 transition over-

ATED over several
to be an overkill.
ates *D* and *F* must
er is probably call-
re chosen for the
HOLDING REG-
oller including the
rve that the grand
lished 14 or 16 pin

ent

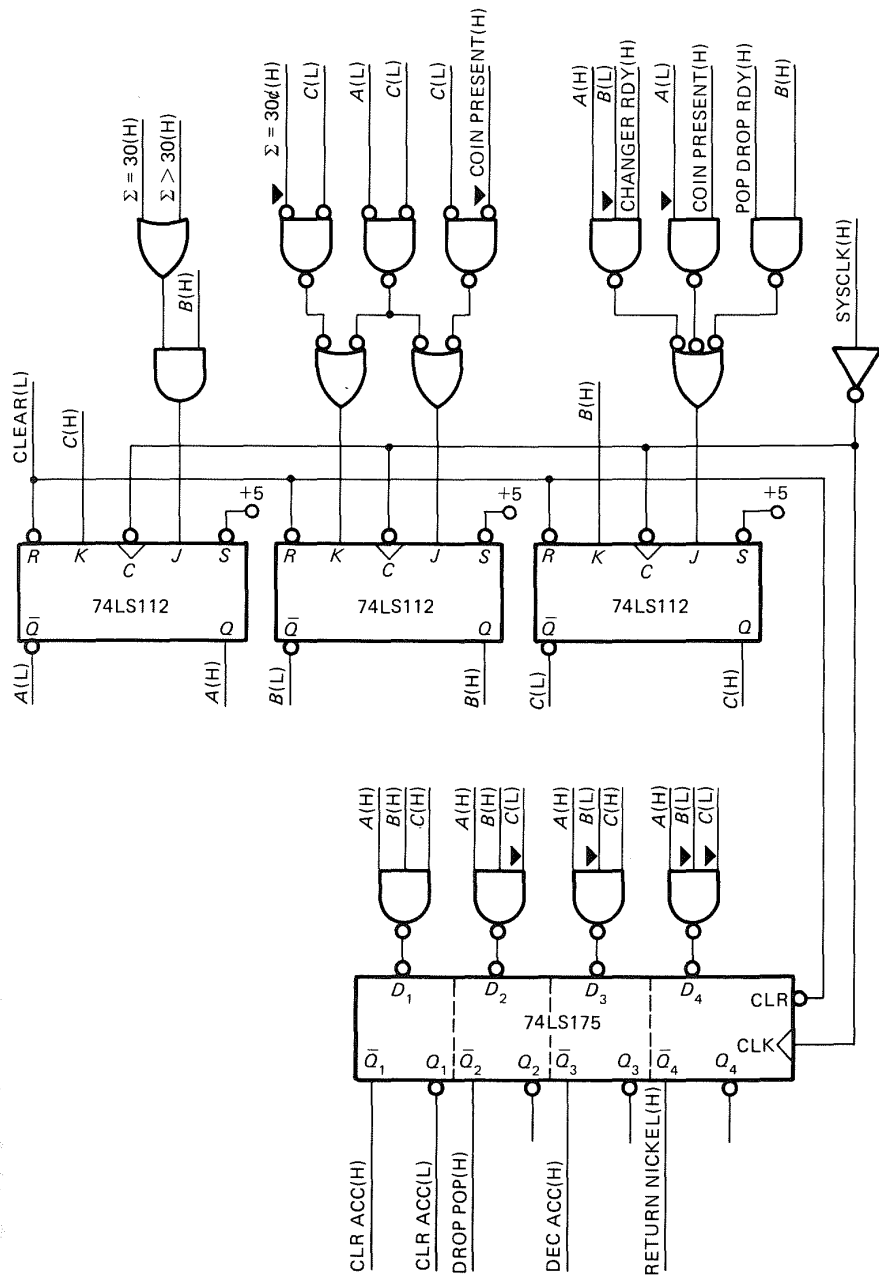


Fig. 7-46. The complete JK implementation of the system controller for the pop vending machine system.