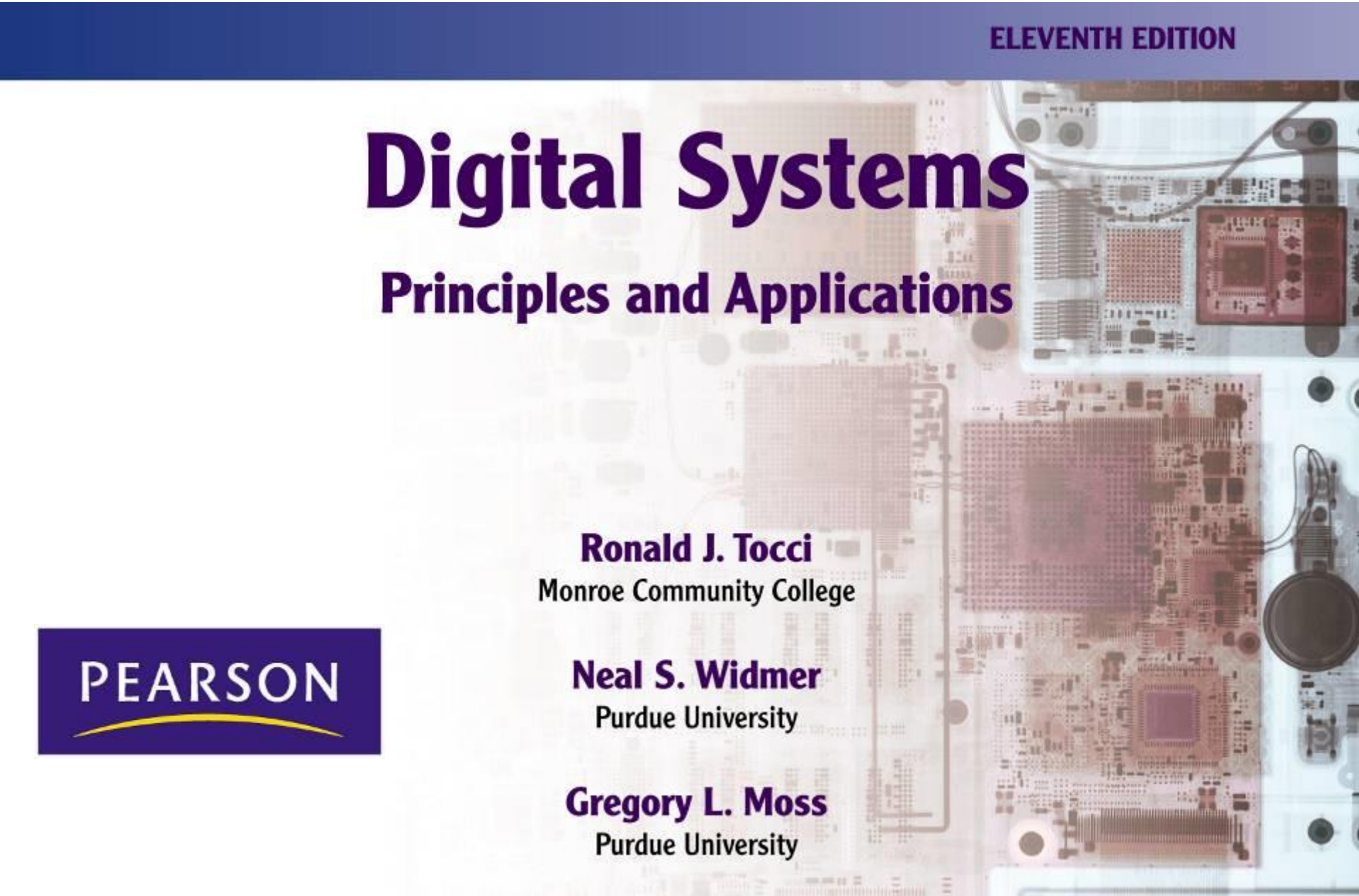

PEARSON

Chapter 10 – Digital System Projects Using HDL



Chapter 10 Objectives

- *Selected areas covered in this chapter*:
  - Analyzing operation of systems made of several components covered earlier in this textbook.
  - Describing an entire project with one HDL file.
  - Describing the process of hierarchical project management.
  - Dividing a project into manageable pieces.
  - Using Quartus II software tools to implement a hierarchical modular project.
  - Developing strategies to test the operation of digital circuits.

10-1 Small Project Management

- Hardware description languages were created for management of large digital systems:
  - Documentation, simulation testing & synthesis of working circuits.
    - The Altera software tools are specifically designed to work with managing projects.
- The general steps are:
  - Overall definition.
  - Strategic planning to break the project into small pieces.
  - Synthesis and testing of each piece.
  - System integration and testing.

10-1 Small Project Management

- Definition of the project:
  - *How many bits of data are needed?*
  - *How many devices are controlled by the outputs?*
  - *What are the names of each input and output?*
  - *Are the inputs and outputs active-HIGH or active-LOW?*
  - *What are the speed requirements?*
  - *Do I fully understand how this should operate?*
  - *What will define successful completion of the project?*

10-1 Small Project Management

- In the strategy for dividing the project into manageable pieces, requirements are:
  - There must be a way to test each piece.
  - They must fit together to make up the whole system.
  - We must know the nature of all the signals that interconnect the pieces.
  - The exact operation of each block must be thoroughly defined and understood.
  - We must have a clear vision of how to make each block work.

- Synthesis and testing:
  - After the subsystem is built, or the HDL code written, a plan for testing must be developed.
    - The simulation must include all possible inputs and responses.
  - If a subsystem is thought to be working perfectly but fails under an untested condition, the entire project can be affected.

- System integration and testing
  - Subsystems are assembled and tested as a unit.
  - This is the stage where unforeseen details surface.
- The importance of time cannot be overstated.
  - Most facets of the project will take 2-3 times longer than thought.

10-2 A Stepper Motor Driver

- A steper motor driver demonstrates a typical application of counters in combination with decoding circuits.
  - The stepper motor and how it operates.

Each state transition involves turning off one coil and simultaneously turning on another coil.

Full-Step	Half-Step	Wave-Drive
1010	1010	
	1000	1000
1001	1001	
	0001	0001
0101	0101	
	0100	0100
0110	0110	
	0010	0010

A summary of stepper motor operating modes:

Mode	M1	M0	Input Signals	Output
0	0	0	Step, direction	Full-step count sequence
1	0	1	Step, direction	Wave-drive count sequence
2	1	0	Step, direction	Half-step count sequence
3	1	1	Four control inputs	Direct drive from control inputs

Inputs

step: rising edge trigger

direction: 0 = upward through table, 1 = downward through table

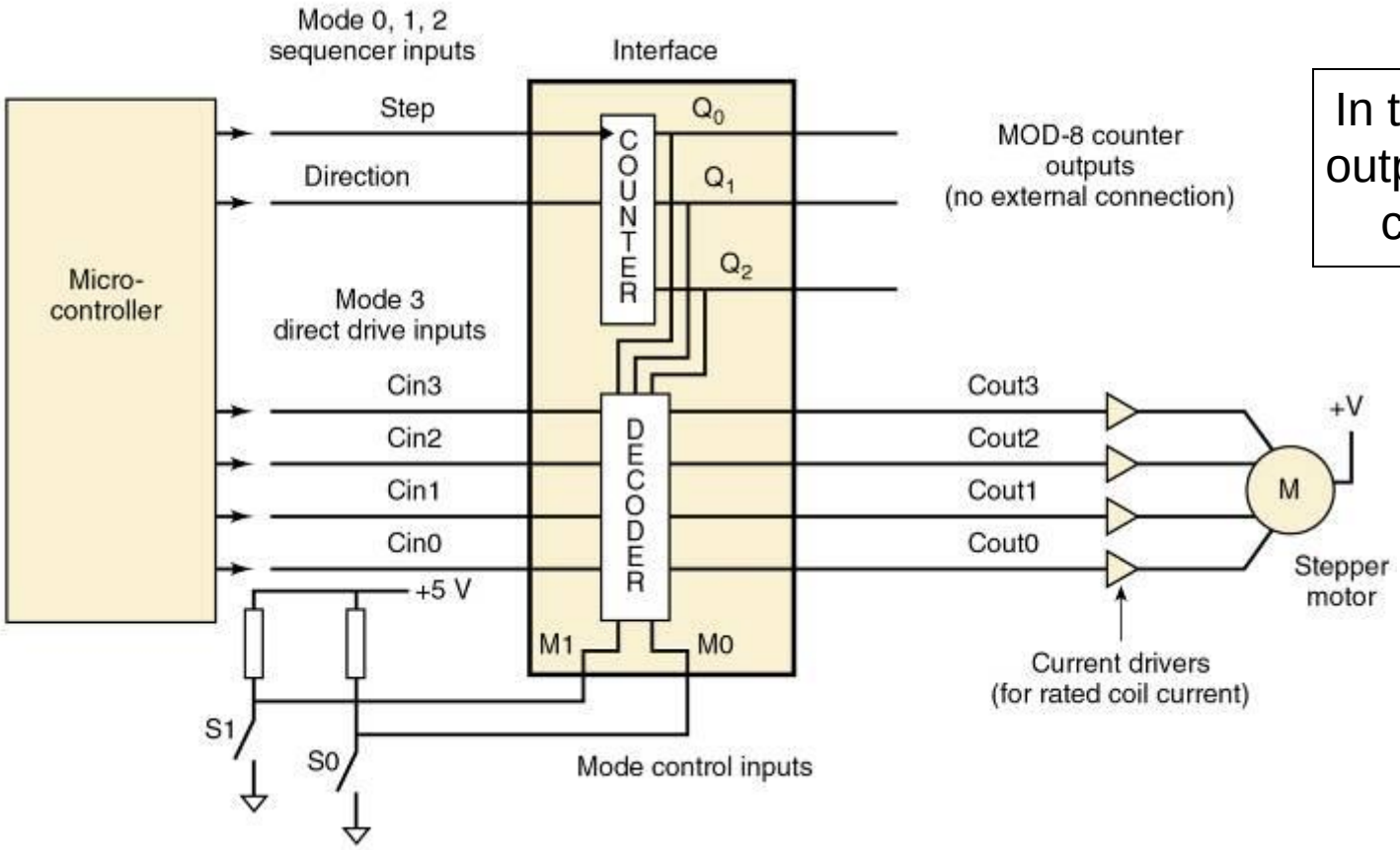
cin0, cin1, cin2, cin3, m1, m0: active-HIGH control inputs

Outputs

cout0, cout1, cout2, cout3: active-HIGH control outputs

10-2 A Stepper Motor Driver

- Strategic planning—key requirements:
  - A sequential counter circuit to control the outputs in three of the modes.



In the last mode, output follows the control input

10-2 A Stepper Motor Driver

- Strategic planning—steps to completion:
  - Build an up/down counter.
  - Make each decoded sequence work with the counter.
  - Make the mode inputs select one decoder sequence and add in the direct drive option.

Success will be realized when the circuit follows the states shown, in either direction, for each mode sequence, and passes the four *cin* signals directly to *cout* in mode 3.

Full-Step	Half-Step	Wave-Drive
1010	1010	
	1000	1000
1001	1001	
	0001	0001
0101	0101	
	0100	0100
0110	0110	
	0010	0010

10-2 A Stepper Motor Driver

- Synthesis and testing
  - Verify the counter counts up, and down through the 8 states.
  - Add one of the decoded outputs and test.
  - Additional count sequences are variations of code already tested.

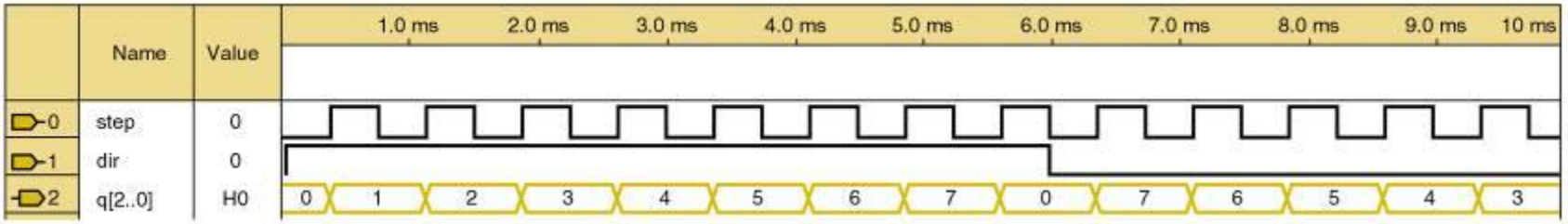
Synthesis and testing

```
SUBDESIGN fig10_2
(
  step, dir      :INPUT;
  q[2..0]        :OUTPUT;
)
VARIABLE
count[2..0]      : DFF;

BEGIN
  count[].clk = step;
  IF dir THEN count[].d = count[].q + 1;
  ELSE        count[].d = count[].q - 1;
  END IF;
  q[] = count[].q;
END;
```

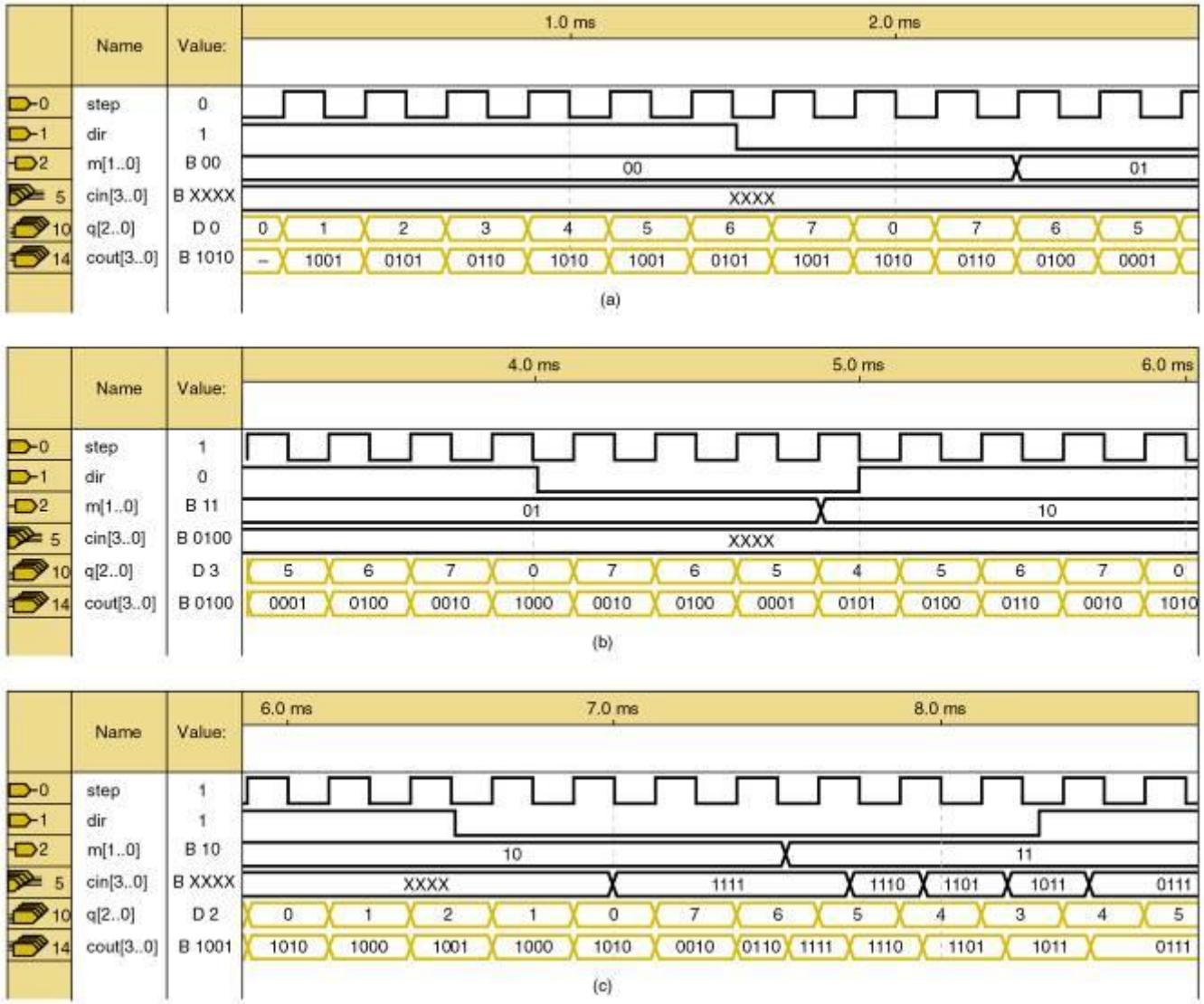
```
ENTITY fig10_3 IS
PORT( step, dir    :IN BIT;
      q            :OUT INTEGER RANGE 0 TO 7);
END fig10_3;

ARCHITECTURE vhdl OF fig10_3 IS
BEGIN
   PROCESS (step)
   VARIABLE count    :INTEGER RANGE 0 TO 7;
   BEGIN
      IF (step'EVENT AND step = '1') THEN
         IF dir = '1' THEN count := count + 1;
         ELSE              count := count - 1;
         END IF;
      END IF;
      q <= count;
   END PROCESS;
END vhdl;
```



10-2 A Stepper Motor Driver

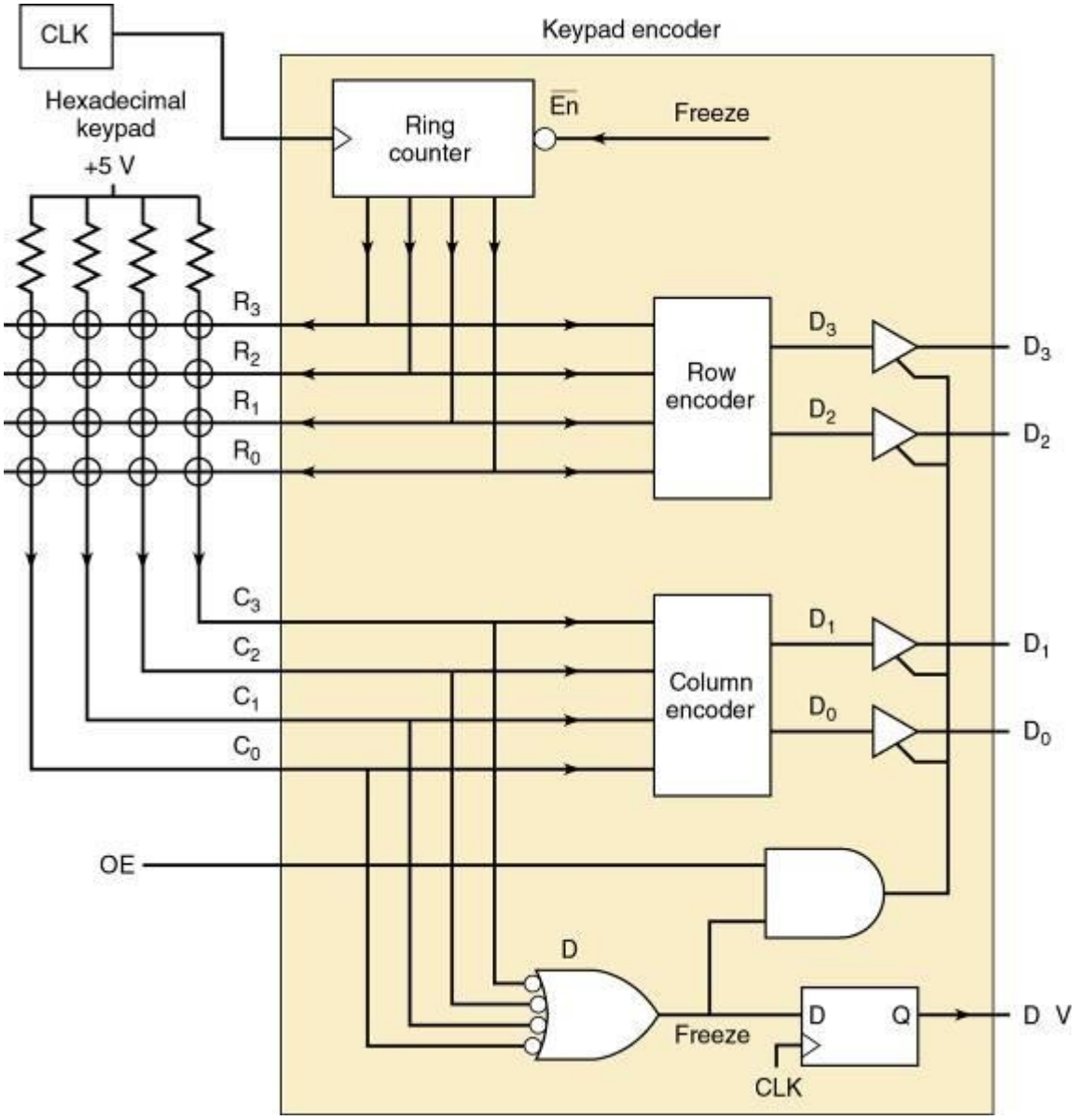
The circuit is working.



10-3 A Keypad Encoder

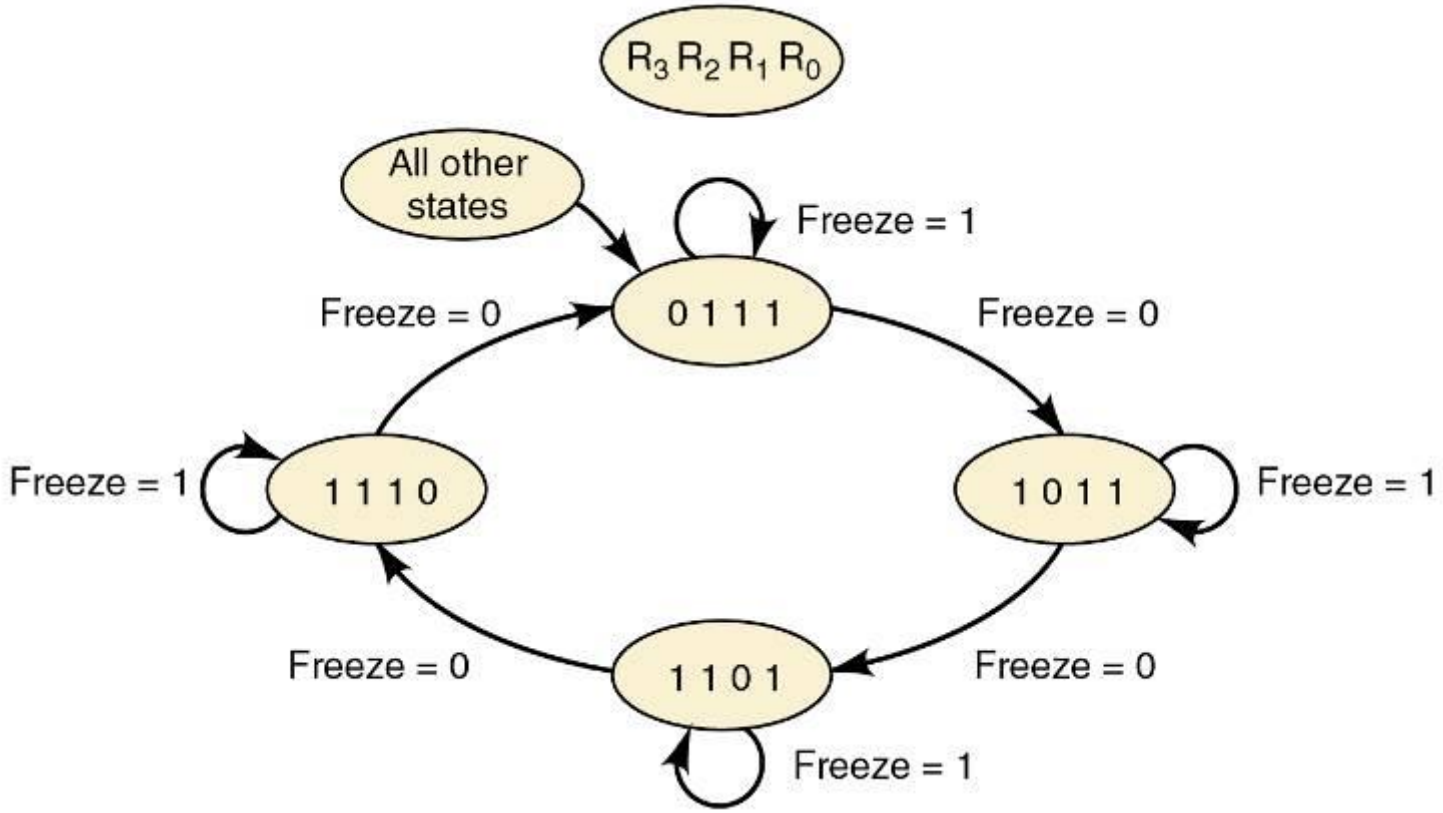
Encode a 16 key keypad to 4-bit binary output.

Because keypads are often interfaced to a computer bus system, encoded outputs should have tristate enables.



10-3 A Keypad Encoder

- Strategic planning—major circuit blocks:
  - A ring counter with active LOW outputs.



10-3 A Keypad Encoder

- Strategic planning—major circuit blocks:
  - Two encoders for the row and column numbers.

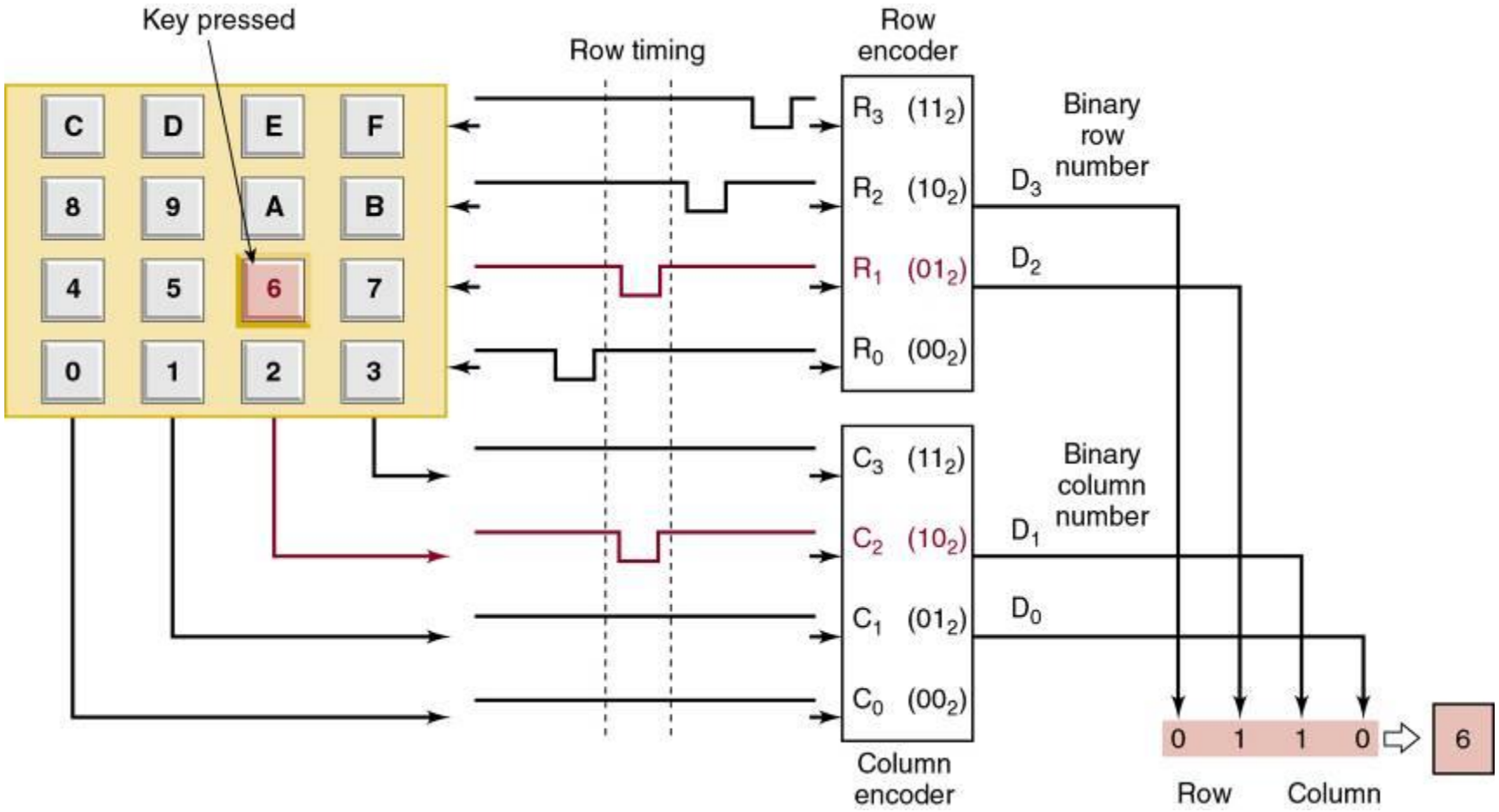
Each state of this counter must be encoded to generate a two-bit binary row number.

Each column value must be encoded to generate a two-bit binary column number.

4	Row drive outputs	R_0–R_3
4	Column read inputs	C_0–C_3
4	Encoded data outputs	D_0–D_3
1	Data available strobe output	*DAV*
1	Tristate enable input	*OE*
1	Clock input	*CLK*

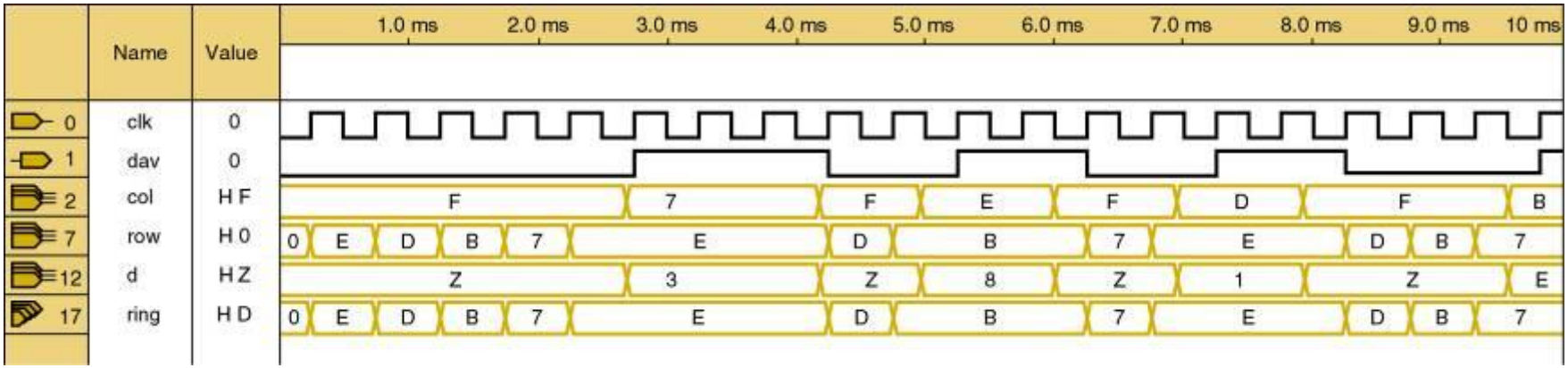
10-3 A Keypad Encoder

- Strategic planning—major circuit blocks:
  - Key press detection and tri-state enable circuits.



10-3 A Keypad Encoder

Simulation of the scanning keypad encoder.



10-4 Digital Clock Project

- A digital clock is a common counter application.
  - Display is given in hour, minutes, seconds.
  - A precise frequency is required.
    - Battery operated devices use a crystal oscillator.
    - 60 Hz ac power line frequency is also used.

Project Specifications.

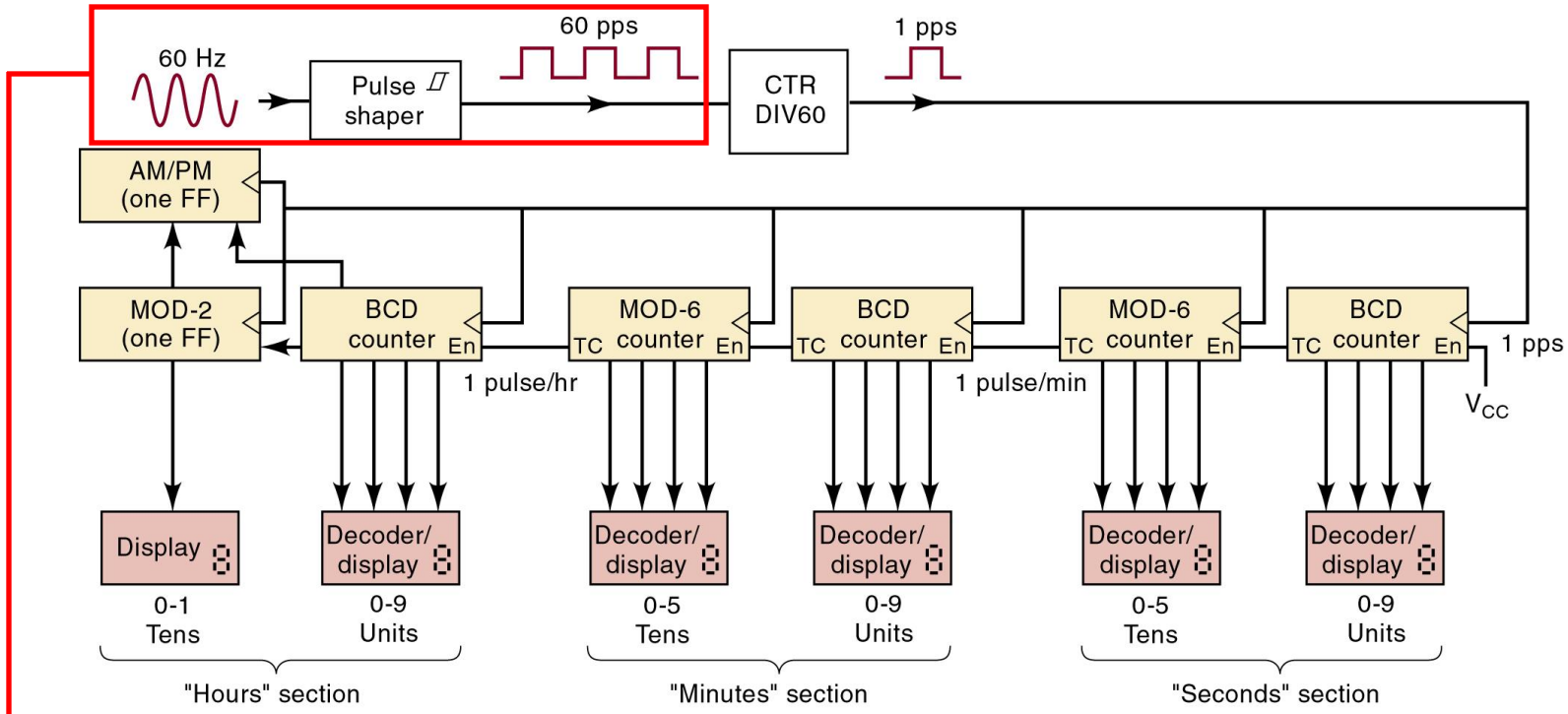
Inputs: 60 pps CMOS compatible waveform (accuracy dependent on line frequency)

Outputs: BCD Hours: 1 bit TENS 4 bits UNITS
BCD Minutes: 3 bits TENS 4 bits UNITS
BCD Seconds: 3 bits TENS 4 bits UNITS
PM indicator

Minutes and Seconds sequence: BCD MOD 60
00–59 (decimal representation of BCD)
Hours sequence BCD MOD 12
01–12 (decimal representation of BCD)
Overall range of display
01:00:00–12:59:59

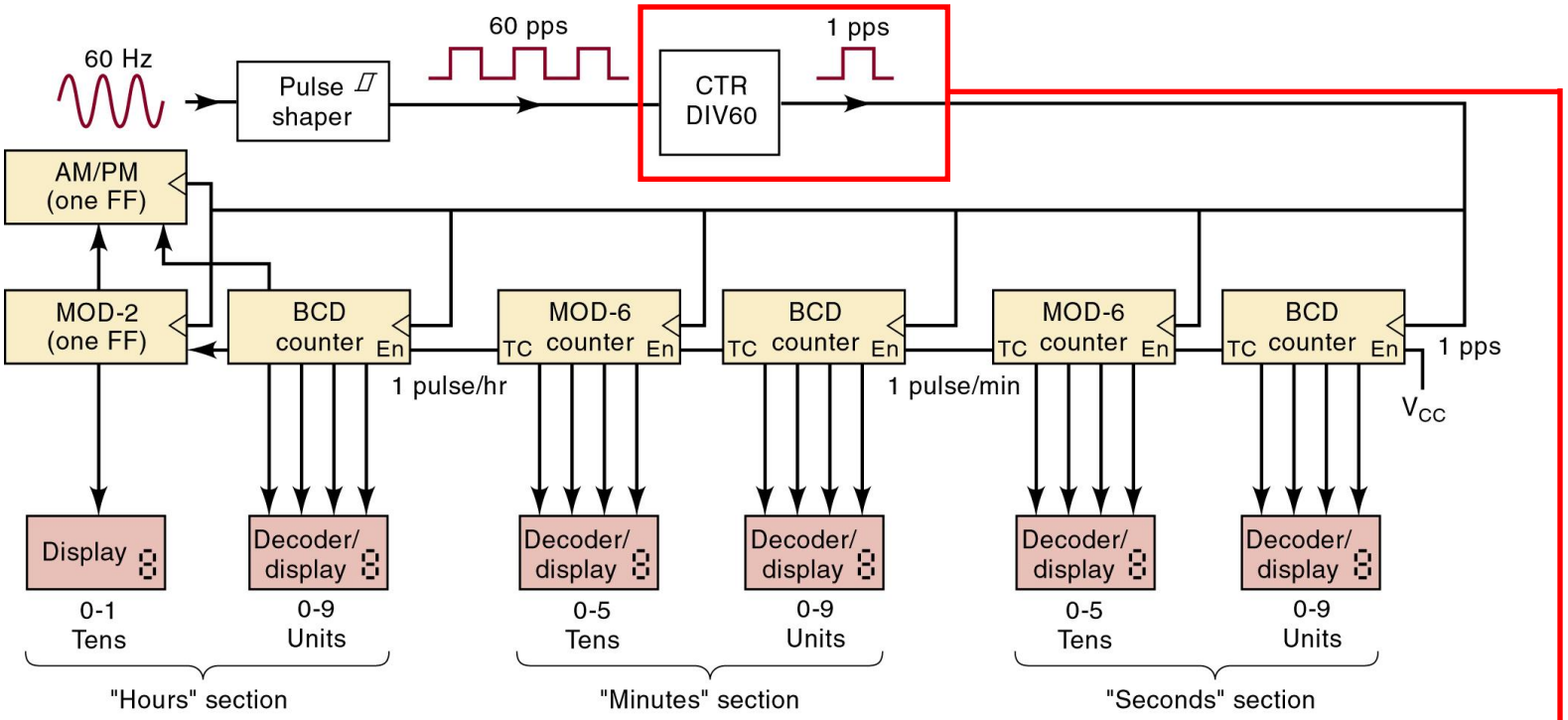
AM/PM indicator toggles at 12:00:00

Block diagram for a clock using the 60 Hz power line frequency.



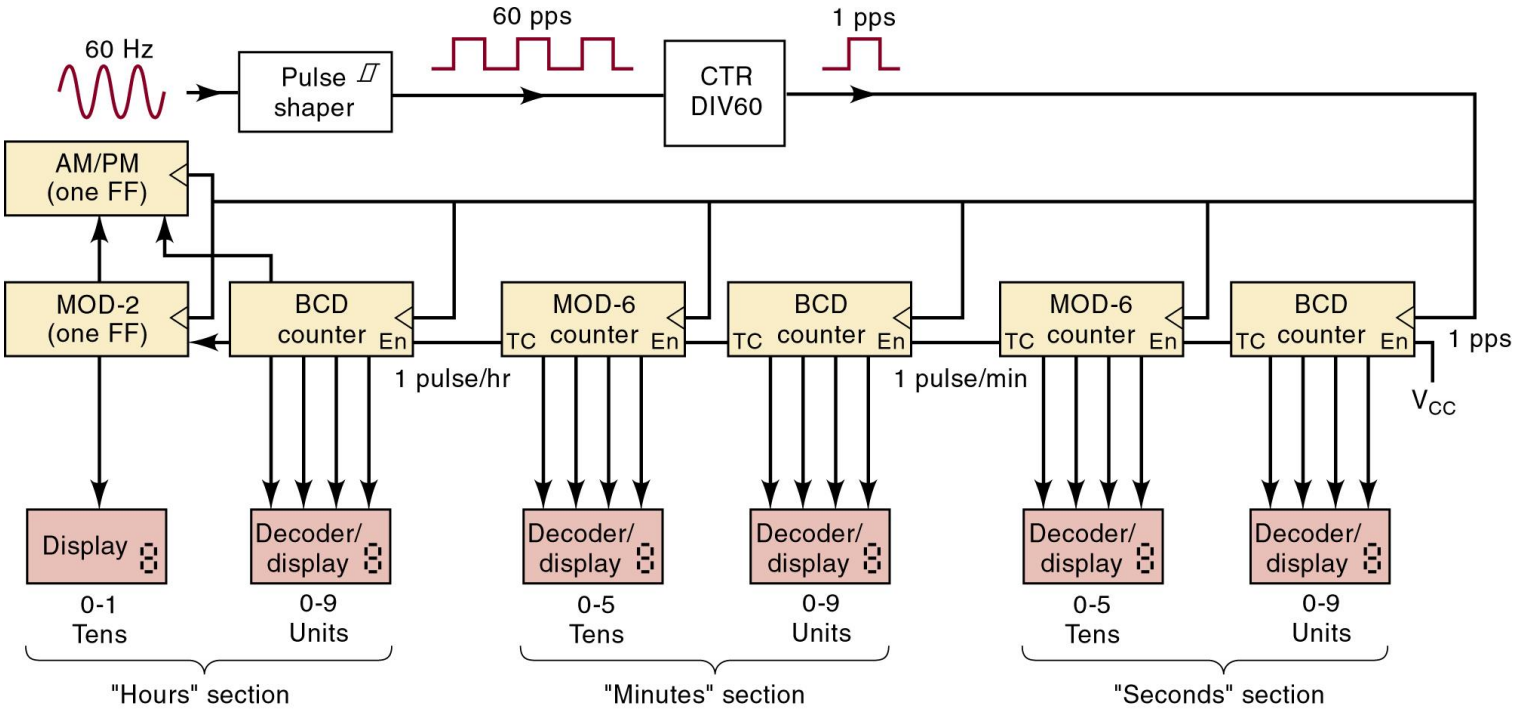
60 Hz is passed through a Schmitt trigger to produce square pulses at 60 pps.

Block diagram for a clock using the 60 Hz power line frequency.



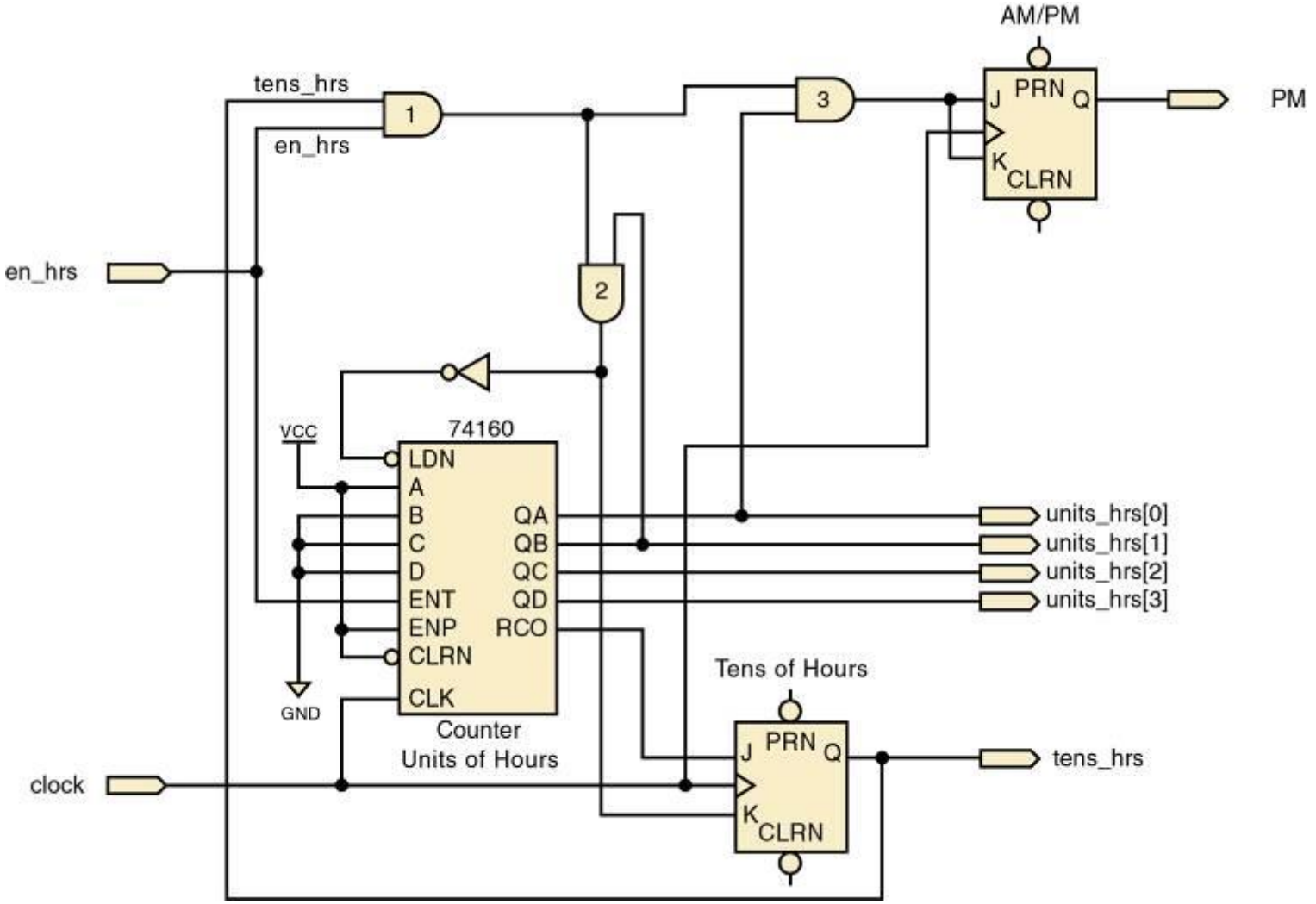
A MOD 60 counter divides 60 to 1 pps.

Block diagram for a clock using the 60 Hz power line frequency.



Stages are synchronously cascaded.

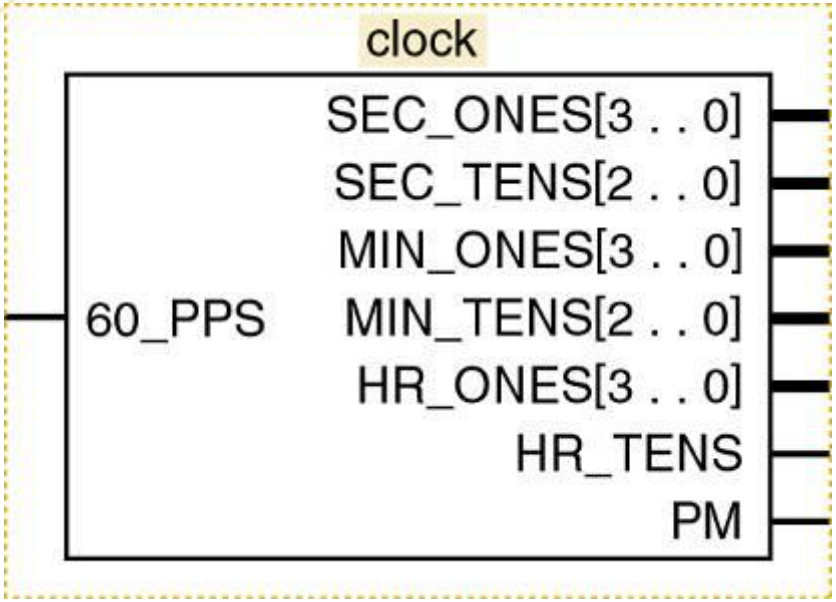
Detail of HOURS section circuit.



10-4 Digital Clock Project

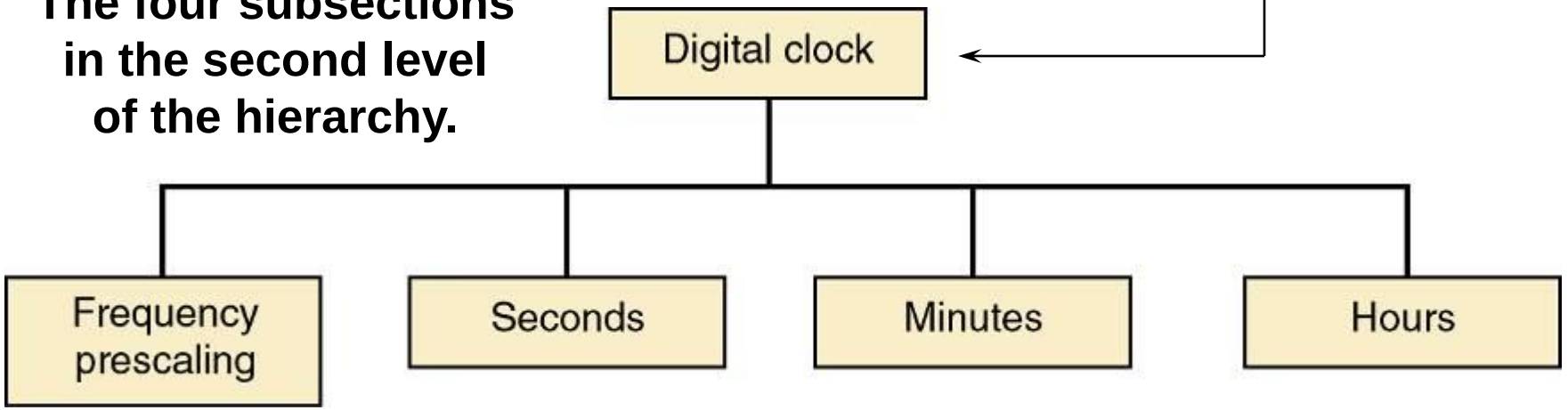
- Large, complex problems go through multiple levels of problem decomposition.
  - Referred to as a **hierarchy**.
- At each level, the interconnections between blocks should be as simple as possible.
  - With clear vision of function, a testing plan, and a watch for common elements that can be reused.

10-4 Digital Clock Project

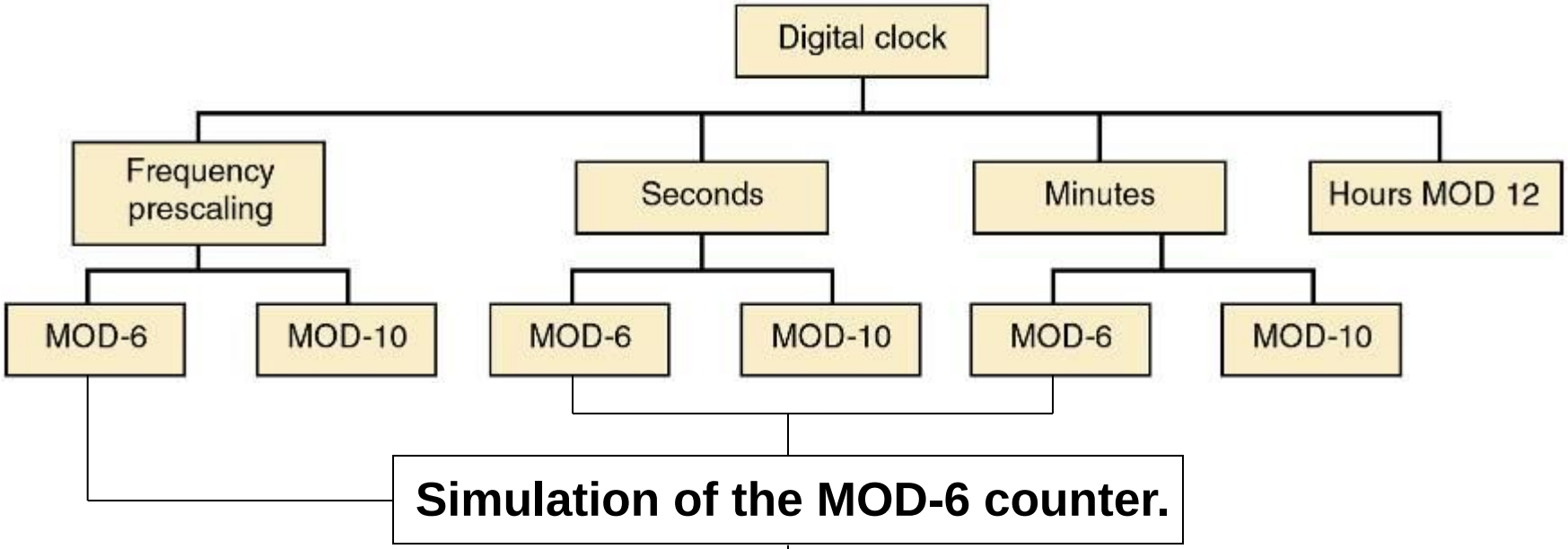


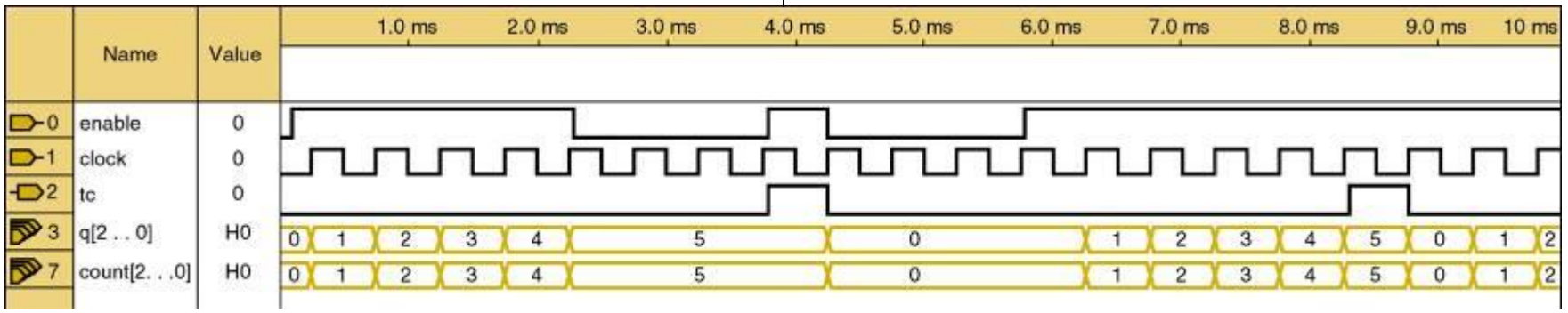
The top level block of the hierarchy.

The four subsections in the second level of the hierarchy.



The complete hierarchy of the clock project.





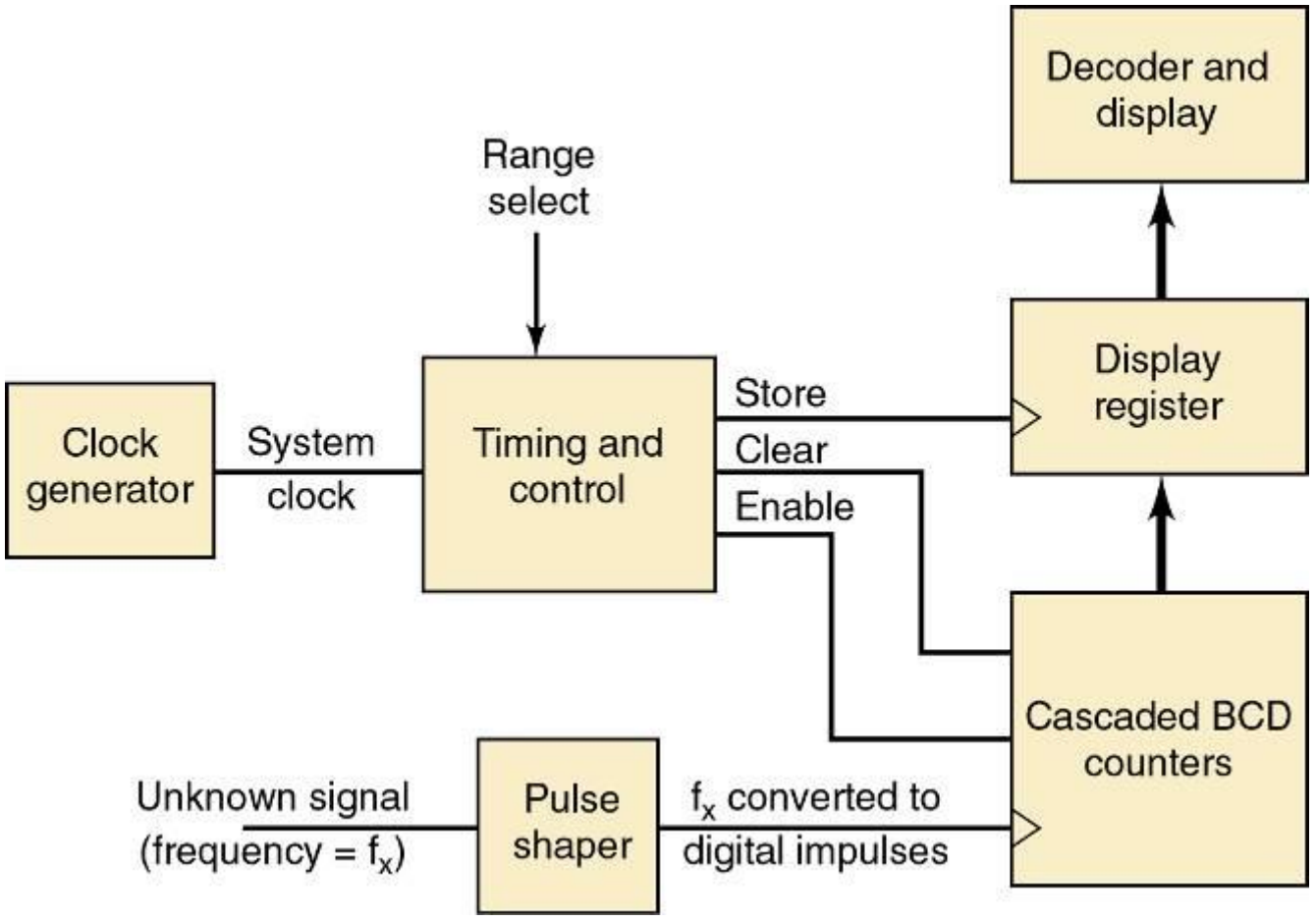
10-6 Frequency Counter Project

- A **frequency counter** is a circuit that can measure & display the frequency of a signal.
  - The frequency of a periodic waveform is the number of cycles per second.
- Shaping each cycle of the frequency into a digital pulse allows a digital circuit to count the cycles.
  - Enabling a count of cycles (pulses) of the incoming waveform during a precisely specified period of time.
    - The **sampling interval**.

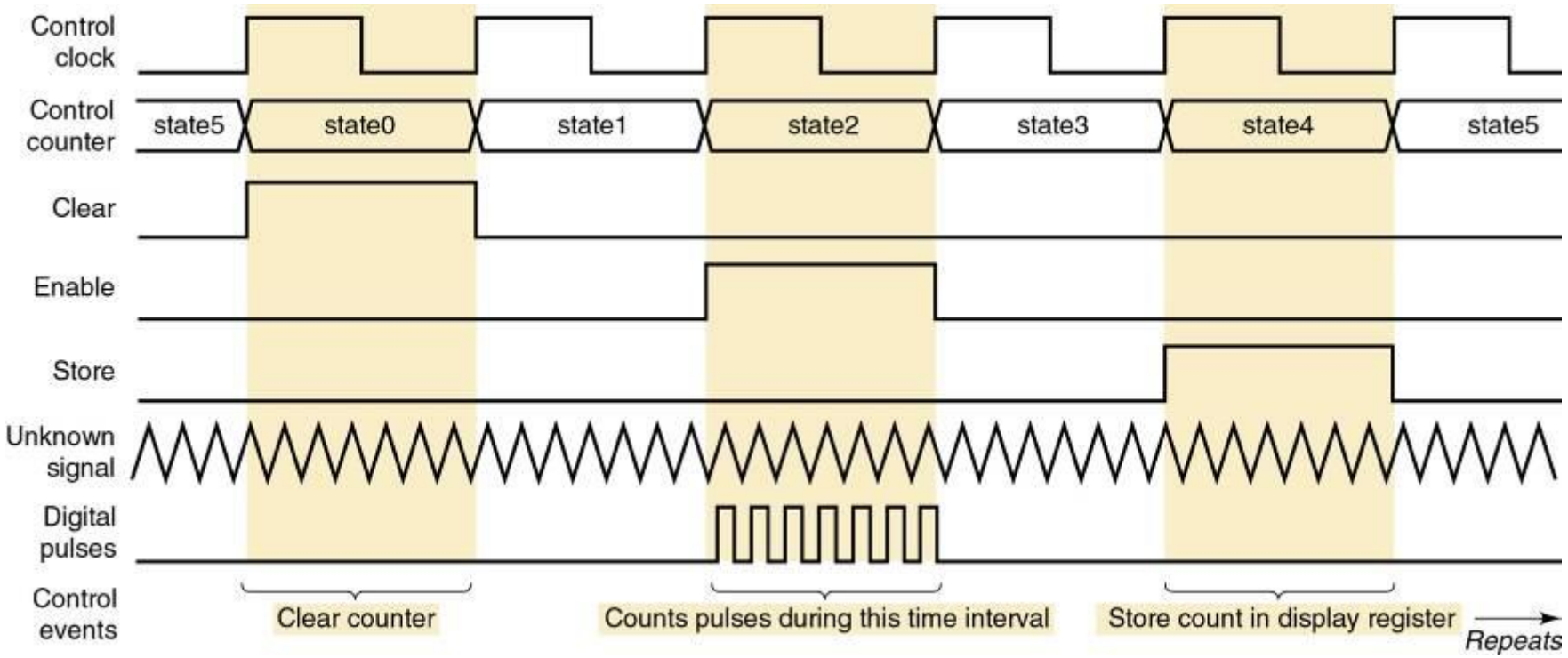
10-6 Frequency Counter Project

- Length of the sampling interval determines the range of frequencies that can be measured.
  - A longer interval provides improved precision for low frequencies—but will overflow at high frequencies.
  - A shorter interval provides less precise measurement of low frequencies.
    - But can measure a much higher maximum frequency without exceeding the upper limit of the counter.

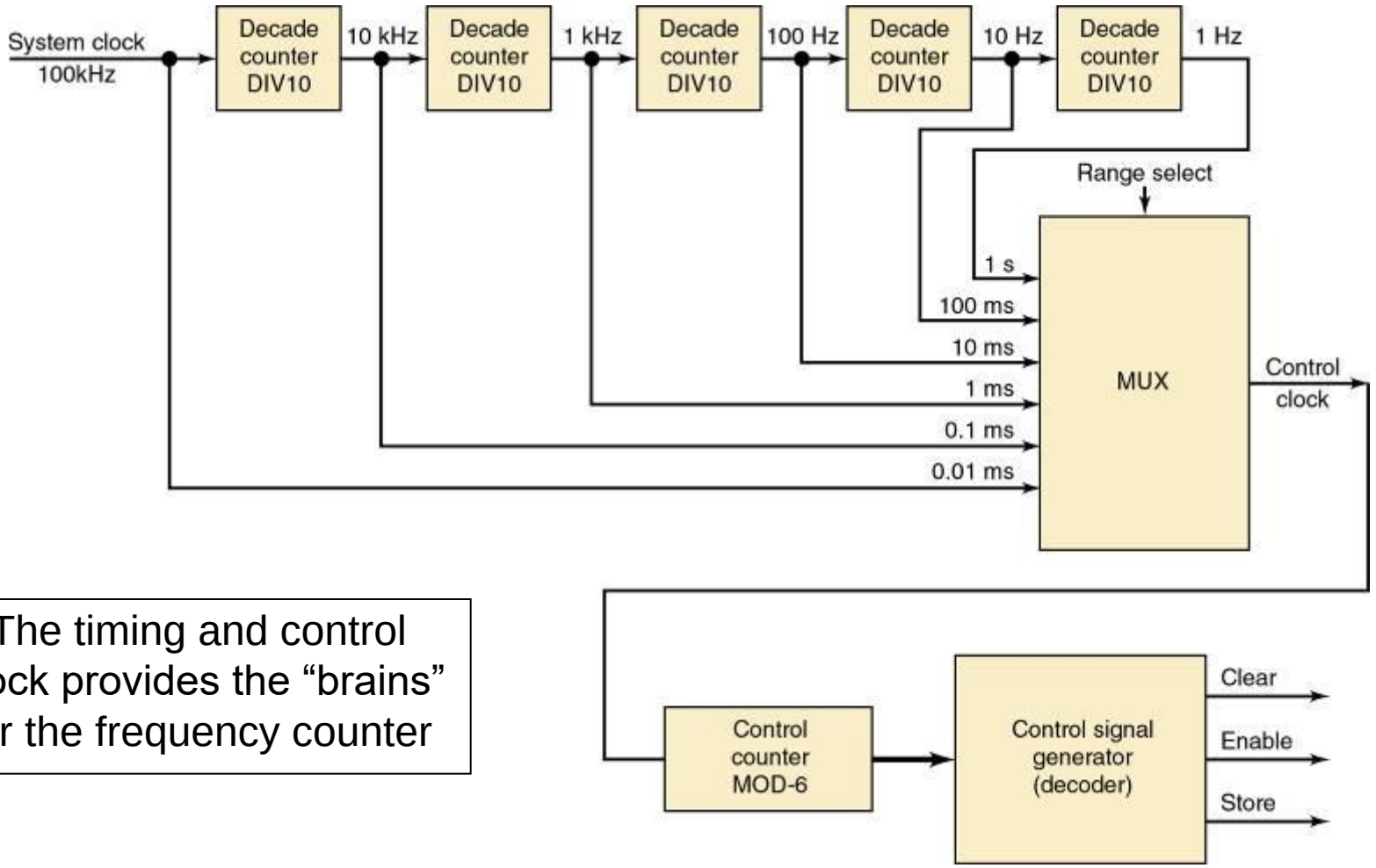
Basic frequency counter block diagram.



Frequency counter timing diagram.



Timing and control block for frequency counter.



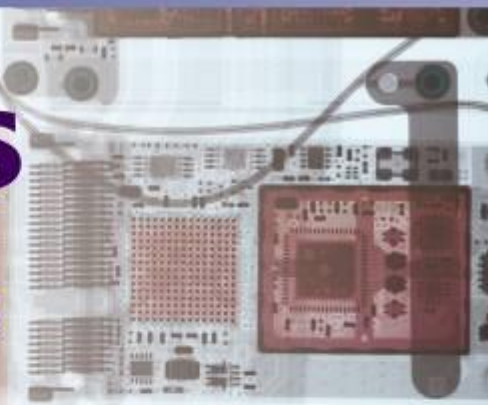
The timing and control block provides the “brains” for the frequency counter

END

ELEVENTH EDITION

Digital Systems

Principles and Applications



Ronald J. Tocci
Monroe Community College

Neal S. Widmer
Purdue University

Gregory L. Moss
Purdue University

PEARSON