Logic and Computer Design Fundamentals

Chapter 7 – Registers and Register Transfers

Part 3 – Control of Register Transfers

Charles Kime

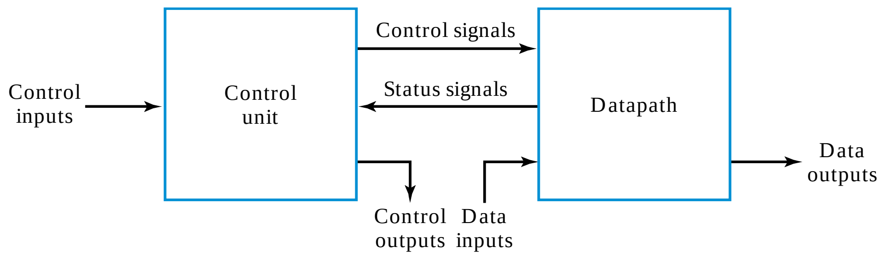
(Hyperlinks are active in View Show mode)

Overview

- **Part 1 – Registers, Microoperations and Implementations**
- **Part 2 – Counters, Register Cells, Buses, & Serial Operations**
- **Part 3 – Control of Register Transfers**
  - **Introduction to register transfer systems**
  - **Register transfer system design procedure**
  - **A design example**
  - **Microprogrammed control**

Introduction to Register Transfer Systems

- **Datapath and Control Unit**



  - **Set of registers, mostly in Datapath with some in Control Unit**
  - **Register transfers performed on registers**
  - **Control that supervises the sequencing of the register transfers**

Programmable and Non-Programmable Systems

- ***Programmable System*** **– a portion of the input consists of a sequence of instructions called a *program*, typically stored in a memory and addressed by a *program counter*. The Control Unit is responsible for fetching and executing these instructions.**
- **Non-programmable System – the control unit does not deal with fetching and executing instructions, but contains all of the information for sequencing register transfers based on inputs and on status bits from the datapath.**
- **Only non-programmable designs are considered here.**

Register Transfer System Design Procedure

- Write a detailed system specification
- Determine all data, control and status input signals, all data, control and status output signals, and registers of the datapath and control unit.
- Find a state machine diagram for the system including register transfers for the datapath and control unit as outputs.
- Determine all internal control and status signals. Use these signals to separate output conditions and actions, including register transfers, from the state machine diagram flow and represent them in tabular form.
- Draw a block diagram of the datapath including all control and status inputs and outputs. Draw a block diagram of the control if it includes register transfer hardware.
- Design any specialized register transfer logic as needed for the datapath and the control.
- Design the control unit logic.
- Verify the correct operation of the combined datapath and control unit. If verification fails, debug the system and verify the changed system.

Design Example – DASHWATCH - Specs

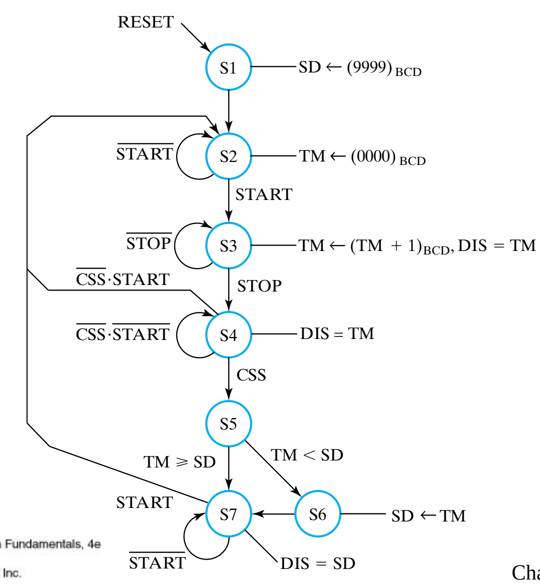
- Very Inexpensive Stop Watch for “dash” runners
- Times intervals to at most 99.99 seconds
- Stopwatch action plus storage of best performance time per session (session ended by turning off power or pushing RESET)
- Inputs START, STOP, CSS (compare and store shortest), RESET
- Registers: 4-digit BCD Counter and 16-bit Parallel Load Register
- Output: 4 digit BCD LCD with decimal point

DASHWATCH Inputs, Outputs, and Registers

- **TABLE 7-15**
  Inputs, Outputs, and Registers of the DashWatch

Symbol	Function	Type
START	Initialize timer to 0 and start timer	Control input
STOP	Stop timer and display timer	Control input
CSS	Compare, store and display shortest dash time	Control input
RESET	Set shortest value to 10011001	Control input
B_1	Digit 1 data vector a, b, c, d, e, f, g to display	Data output vector
B_0	Digit 0 data vector a, b, c, d, e, f, g to display	Data output vector
DP	Decimal point to display (= 1)	Data output
B_{-1}	Digit –1 data vector a, b, c, d, e, f, g to display	Data output vector
B_{-2}	Digit –2 data vector a, b, c, d, e, f, g to display	Data output vector
B	The 29-bit display input vector (B_1, B_0, DP, B_{-1}, B_{-2})	Data output vector
TM	4-Digit BCD counter	16-Bit register
SD	Parallel load register	16-Bit register

DASHWATCH State Machine Diagram with Register Transfer Outputs



State Machine Diagram Design

- Specify only Moore outputs (no particular reason)
- S1: Reset state - in this state, initialize SD to 1001100110011001 (99.99), the maximum possible dash time.
- S2: Because of Moore output spec, S1 cannot be used for this state since SD is not to be initialized again to 99.99 after having passed through states S4 or S7. TM is initialized to $(0000)_{BCD}$ for next dash.
- S3: State during dash. Entered with START and exited with STOP. While in state, 1 (0.01 seconds) is added to TM for each clock pulse. (Clock frequency is 100 Hz), and DIS shows TM value.
- S4: Decision state whether to Compare, Store, and display Shortest dash time, or to continue to display TM. Also START begins new dash.
- S5: State for comparison of TM to SD.
- S6: State for loading TM into SD if TM is smaller.
- S7: State for START to begin new dash and display of SD as shortest dash time.

DASHWATCH Output Control/Status Table

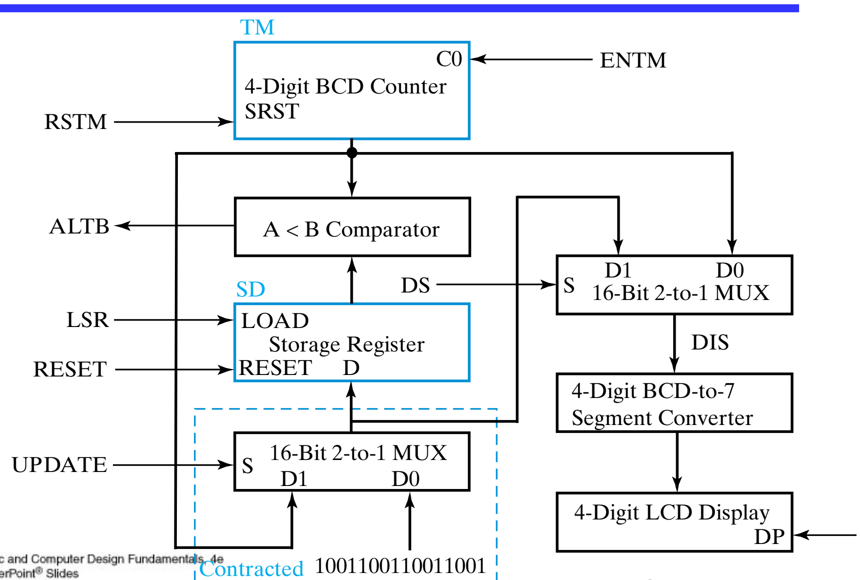
TABLE 7-16
Datapath Output Actions and Status Generation with Control and Status Signals

Action or Status	Control or Status Signals	Meaning for Values 1 and 0
TM $(0000)_{BCD}$	RSTM	1: Reset TM to 0 (synchronous reset) 0: No reset of TM
TM $(TM + 1)_{BCD}$	ENTM	1: BCD count up TM by 1, 0: hold TM value
SD $(9999)_{BCD}$	UPDATE LSR	0: Select 1001100110011001 for loading SD 1: Enable load SD, 0: disable load SD
SD TM	UPDATE LSR	1: Select TM for loading SD Same as above
DIS = TM DIS = SD	DS	0: Select TM for DIS 1: Select SD for DIS
TM < SD TM SD	ALTB	1: TM less than SD 0: TM greater than or equal to SD

Determination of Internal Control/Status Signals

- **TM – Timer**
  - **Reset to 0000: RSTM**
  - **Enable to Count Up: ENTM**
- **SD – Shortest Dash**
  - **Load SD: LSR = 1;**
  - **Select input 9999: UPDATE = 0**
  - **Select input TM: UPDATE = 1**
- **DIS – Display (B_1, B_0, DP, B_{-1}, B_{-2})**
  - **Select input TM: DS = 0**
  - **Select input SD: DS = 1**
- **Compare TM and SD (Status)**
  - **TM < SD: ALTB = 1**
  - **TM ≥ SD: ALTB = 0**

DASHWATCH Datapath



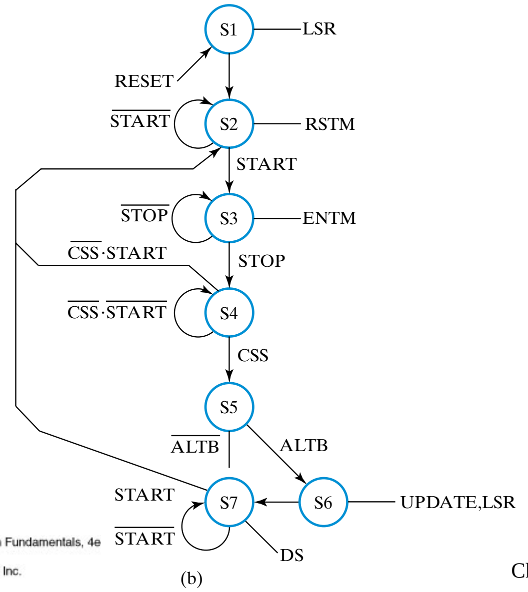
DASHWATCH – Datapath Development

- TM: 4-digit BCD Counter with Synchronous Reset
  - Based on previous BCD adder digit design
  - synchronous reset SRST added
  - SRST = RSTM
  - C0 (Incoming carry) = ENTM
- A < B Comparator
  - Compares TM to SD
  - Designed as left-to-right iterative cell array with output C0
- SD: Standard 16-bit parallel load register
  - LOAD = LSR
  - Contracted standard 2-way, 16-bit multiplexer used to select between 9999_{BCD} and TM as parallel load input D
  - S = UPDATE

DASHWATCH – Datapath Development – Display Logic

- 2-way 16-bit multiplexer
  - Selects between TM and SD
  - S = DS
- 4-digit BCD-to-7 Segment Converter
  - Uses previous design
- 4-digit 7-Segment Display with Decimal Point
  - 2-digit fractional part
  - Decimal Point control = DP
  - DP = 1

DASHWATCH – SMD with Control Signal Outputs Replacing Register Transfers



DASHWATCH – FF Input Equations

- **One-Hot State Assignment – 7 bits**
- **State S1 entered only by using asynchronous RESET**

$$D_{S1} = S1(t+1) = 0$$

$$D_{S2} = S2(t+1) = S1 + S2 \cdot \overline{START} + S4 \cdot \overline{CSS} \cdot START + S7 \cdot START$$

$$D_{S3} = S3(t+1) = S2 \cdot START + S3 \cdot \overline{STOP}$$

$$D_{S4} = S4(t+1) = S3 \cdot STOP + S4 \cdot \overline{CSS} \cdot \overline{START}$$

$$D_{S5} = S5(t+1) = S4 \cdot CSS$$

$$D_{S6} = S5 \cdot ALTB$$

$$D_{S7} = S7(t+1) = S5 \cdot \overline{ALTB} + S6 + S7 \cdot \overline{START}$$

DASHWATCH – Output Equations

$$LSR = S1 + S6$$

$$RSTM = S2$$

$$ENTM = S3$$

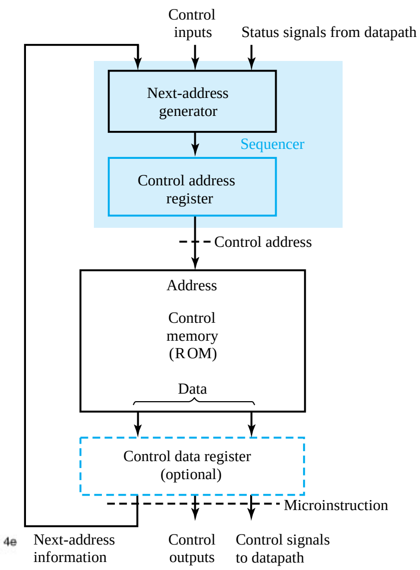
$$UPDATE = S6$$

$$DS = S7$$

Microprogrammed Control

- ***Microprogrammed Control*** **— a control unit with binary control values stored as <u>words</u> in <u>memory.</u>**
- ***Microinstructions*** **— words in the control memory.**
- ***Microprogram*** **— a sequence of microinstructions.**
- ***Control Memory*** **— RAM or ROM memory holding the microinstructions.**
- ***Writeable Control Memory*** **— RAM Memory into which microinstructions may be written**

Microprogrammed Control (continued)



Terms of Use

- **All (or portions) of this material © 2008 by Pearson Education, Inc.**
- **Permission is given to incorporate this material or adaptations thereof into classroom presentations and handouts to instructors in courses adopting the latest edition of Logic and Computer Design Fundamentals as the course textbook.**
- **These materials or adaptations thereof are not to be sold or otherwise offered for consideration.**
- **This Terms of Use slide or page is to be included within the original materials or any adaptations thereof.**