ECE380 Digital Logic

State Minimization

Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 32-1

State minimization

- For simple FSMs, it is easy to see from the state diagram that the number of states used is the minimum possible
 - FSMs for counters are good examples
- For more complex FSMs, it is likely that an initial state diagram may have more states than are necessary to perform a required function
- Minimizing states is of interest because fewer states implies fewer flip-flops to implement the circuit
 - Complexity of combinational logic may also be reduced
- Instead of trying to show which states are equivalent, it is often easier to show which states are definitely **not** equivalent
 - This can be exploited to define a minimization procedure

Electrical & Computer Engineering

State equivalence

- **Definition**: Two states S_i and S_j are equivalent if and only if for every possible input sequence, the same output sequence will be produced regardless of whether S_i or S_j is the initial state
- If an input w=0 is applied to an FSM in state S_i and the FSM transitions to state S_u, then S_u is termed a *O-successor* of S_i
- Similarly, if w=1 and the FSM transitions to S_y , then S_y is a **1-successor** of S_i
- The successors of S_i are its k-successors
- With one input, k can only be 0 or 1, but if there are multiple inputs then k represents all the valuations of the inputs

Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 32-3

Partitioning minimization

- From the equivalence definition, if S_i and S_j are equivalent then their corresponding *k-successors* are also equivalent
- Using this, we can construct a minimization procedure that involves considering the states of the machine as a set and then breaking that set into partitions that comprise subsets that are definitely not equivalent
- **Definition**: A partition consists of one or more blocks, where each block comprises a subset of states that may be equivalent, but the states in one block are definitely not equivalent to the states in other blocks

Electrical & Computer Engineering

Partition minimization example

Consider the following state table

Present state	Next state		Output
	w = 0	w = 1	ž
Α	В	С	1
В	D	F	1
C	F	Е	0
D	В	G	1
E	F	С	0
F	E	D	0
G	F	G	0

An initial partition contains all the states in a single block P_1 =(ABCDEFG)

Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 32-5

Partition minimization example

- The next partition separates the states that have different outputs
 - $-P_2=(ABD)(CEFG)$
- Now consider all 0- and 1-successors of the states in each block
 - For (ABD), the 0-successors are (BDB)
 - Since these are all in the same block we must still consider A, B, and D equivalent
 - The 1-successors of (ABD) are (CFG)
 - \bullet We must still consider A, B, and D equivalent
 - Now consider the (CEFG) block

Electrical & Computer Engineering

Partition minimization example

- P₂=(ABD)(CEFG)
- For (CEFG), the 0-successors are (FEFF) which are all in the same block in P₂
 - C, E, F, and G must still be considered equivalent
- The 1-successors of (CEFG) are (ECDG)
 - Since these are not in the same block in P₂ then at least one of the states in (CEFG) is not equivalent to the others
 - State F must be different from C, E and G because its 1successor, D, is in a different block than E, C, and G
- Therefore, P₃=(ABD)(CEG)(F)
- At this point, we know that state F is unique since it is in a block by itself

Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 32-7

Partition minimization example

- P₃=(ABD)(CEG)(F)
- The process repeats yielding the following
 - 0-successors of (ABD) are (BDB)
 - A, B and D are still considered equivalent
 - 1-successors of (ABD) are (CFG), which are not in the same block
 - B cannot be equivalent to A and D since F is in a different block than C and G
 - The 0- and 1-successors of (CEG) are (FFF) and (ECG)
 - C, E, and G must still be considered equivalent
- Thus, P₄=(AD)(B)(CEG)(F)

Electrical & Computer Engineering

Partition minimization example

- If we repeat the process to check the 0- and 1-successors of the blocks (AD) and (CEG), we find that
 - $-P_5=(AD)(B)(CEG)(F)$
- Since P₅=P₄ and no new blocks are generated, it follows that states in each block are equivalent
 - A and D are equivalent
 - C, E, and G are equivalent

Electrical & Computer Engineering

Dr. D. J. Jackson Lecture 32-9

Partition minimization example

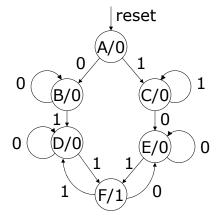
- The state table can be rewritten, removing the rows for D, E and G and replacing all occurrences of D with A and all occurrences of E or G with C
- The resulting state table is as follows

Present	Nextstate		Output
state	w = 0	w = 1	z
Α	В	С	1
В	Α	F	1
C	F	С	0
F	С	Α	0

Electrical & Computer Engineering

Design Example

• Determine which states (if any) are equivalent in the following Moore state diagram



Electrical & Computer Engineering