

3.5 Finite state machines

Name: _____ Class: _____ Date: _____

Total: 21 marks**KEY WORDS** finite state machine

• state diagram

• Moore machine

• Mealy machine

Objective

Design finite state machines from a plain-English behaviour: state diagram → state table → encodings → excitation equations, choosing Moore or Mealy deliberately, and merge redundant states.

1 Key knowledge and worked examples

■ States, transitions, outputs

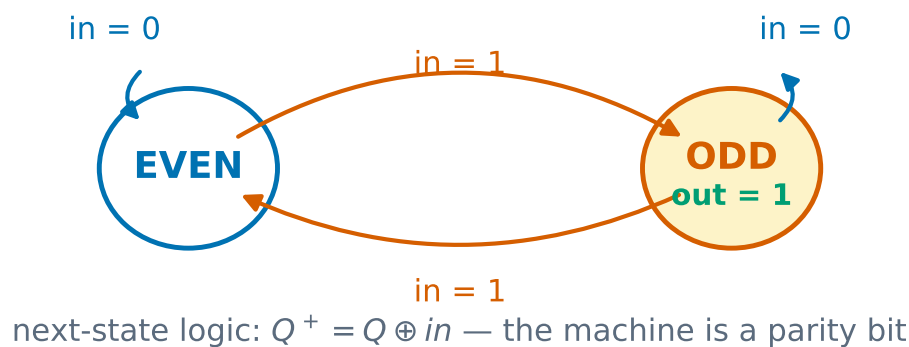
A **finite state machine** (FSM) = states + transitions + outputs.

A **state diagram** draws states as bubbles, transitions as labelled arrows (input/output or input per arrow). **Moore** outputs depend on state only;

Mealy outputs depend on state AND inputs – faster (same-cycle) but glitch-prone.

Design procedure: (1) state diagram from the behaviour; (2) **state table** (next state + output per state/input); (3) binary encoding; (4) excitation table per flip-flop; (5) minimise next-state logic (Karnaugh maps).

The parity detector: two states, one flip-flop



■ Worked example: the parity detector

Detect odd number of 1s. Two states: EVEN, ODD.

state	in=0 →	in=1 →	output
EVEN	EVEN	ODD	0
ODD	ODD	EVEN	1

Trace on 1, 0, 1, 1: EVEN → ODD → ODD → EVEN → ODD. Output after the fourth bit = 1: three 1s seen (odd ✓). Moore encoding: 1 flip-flop ($Q = 1 \Leftrightarrow \text{ODD}$); next-state equation: $Q^+ = Q \oplus in$ – the machine is one XOR and a flip-flop, which the trace confirms: $0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0, 0 \oplus 1 = 1$ ✓.

■ Minimisation: merge equivalent states

Two states are equivalent if their outputs match AND their next states (per input) are equivalent. **Worked example:** a machine with states S2, S3 that both output 1 and both transition to S1 on 0, S0 on 1 – S2 and S3 are equivalent; merging them costs nothing in behaviour and saves a state (possibly a flip-flop).

2 Practice

2.1 For the parity FSM: which state follows ODD on input 1, and what is the output? [2]

2.2 Name the one structural difference between Moore and Mealy outputs, and one consequence of each. [3]

2.3 A two-state FSM has states A, B: both output 0, and both go to B on input 0 and to A on input 1. Are A and B equivalent? Why? [3]

3 Exam-style questions

3.1 Design a Moore FSM that raises its output for exactly one clock after every $1 \rightarrow 0$ input fall (a fall detector; output otherwise 0).

Draw (describe) the state diagram with states named for their meaning, write the state table, encode with one flip-flop, and derive the output equation. Trace the input 1, 1, 0, 0, 1, 0 giving the output after each bit. [7]

3.2 A vending machine's coin slot accepts only 10p and 20p coins; a 30p item vends and resets to zero credit; overpayment is held (credit 30 or 40 stays "paid").

Model the credit states and transitions (coin inputs only), give the state table, and state how many flip-flops the encoded machine needs and why. Then explain what changes if a "coin return" input must refund exact credit. [6]
