1.4 DATA TYPES & STRUCTURES · 1.4.3

Adders & D-type flip-flops

Logic that adds binary, and logic that remembers a bit. Spec 1.4.3(e).

01 Half adder

Adds	Two bits A, B → sum S, carry C.
Sum	**S = A ⊕ B** (XOR).
Carry	**C = A · B** (AND).
Gates	1 XOR + 1 AND. No carry-in.

02 Full adder

Adds	A, B and **Cin** → S, Cout.
Sum	**S = A ⊕ B ⊕ Cin**
Cout	**(A·B) + (Cin·(A⊕B))**
Build	Two half adders + OR; chain for multi-bit.

03 D-type flip-flop

Stores	**One bit** of data.
Inputs	**D** (data) and a **clock**.
Outputs	**Q** and **¬Q**.
How	On the **clock edge**, Q takes the value of D, then **holds** it.
Use	Building block of **registers & memory**; kept in step by the clock.

FINAL PASS BEFORE THE EXAM

Rapid exam tips

Six slips on adder/flip-flop questions.

01

Half-adder sum = **XOR** (not OR); carry = **AND**.

02

A **full** adder has a third input — the **carry-in**.

03

Full adder = **two half adders + OR**; chain them for multi-bit numbers.

04

Flip-flop inputs = **D + clock**; outputs = **Q and ¬Q**.

05

Q captures D on the **clock edge**, then **holds** until the next tick.

06

Adders **compute**; flip-flops **store**. Different jobs.