Grade 12 A/L ICT · Lesson 4

4.4 Half Adder, Full Adder and Flip-Flops: CPU Blocks and Memory Cells

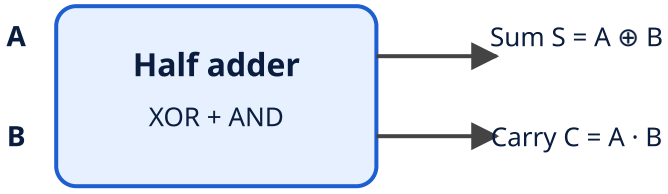
Syllabus 4.4. The CPU adds numbers with **adder** circuits, and memory stores bits with **flip-flops**. In this part you build the truth tables and Boolean expressions for a **half adder** and a **full adder**, see how full adders chain into an ALU, and learn how a **feedback loop** in a **flip-flop** remembers one bit – the building block of registers and RAM.

1. Building blocks of the CPU

Arithmetic in the ALU starts with adding two bits. A **half adder** adds two bits and makes a sum and a carry. A **full adder** also takes a **carry-in** from the previous column, so you can chain eight full adders to add two 8-bit numbers.

2. Half adder

Inputs: A, B. Outputs: **Sum S** and **Carry C**. Looking at binary addition: 0+0=00, 0+1=01, 1+0=01, 1+1=10. So S is 1 when the inputs differ → **S = A ⊕ B**. C is 1 only when both are 1 → **C = A · B**.



Half adder: XOR for Sum, AND for Carry

A	B	Sum S	Carry C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

🧱 **Worked example: For A = 1, B = 1 find S and C of a half adder.**
S = 1 ⊕ 1 = **0**. C = 1 · 1 = **1**. (Binary 1 + 1 = 10_2.)

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

💡 A half adder has **no Cin**, so it can only add the least-significant bit of a multi-bit number. Every higher bit needs a full adder.

3. Full adder

Inputs: A, B, **Cin**. Outputs: **Sum S** and **Cout**. S is 1 when an **odd** number of the three inputs is 1 → **S = A ⊕ B ⊕ Cin**. Cout is 1 when **at least two** inputs are 1 → **Cout = A·B + B·Cin + A·Cin** (the majority function from 4.3).



S = A ⊕ B ⊕ Cin Cout = majority(A,B,Cin)

Full adder with Cin and Cout

A	B	Cin	Sum S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

🧮 **Worked example: For A = 1, B = 0, Cin = 1 find S and Cout.**
S = 1 ⊕ 0 ⊕ 1 = **0**. Cout = majority = **1**. ($1+0+1 = 10_2$.)

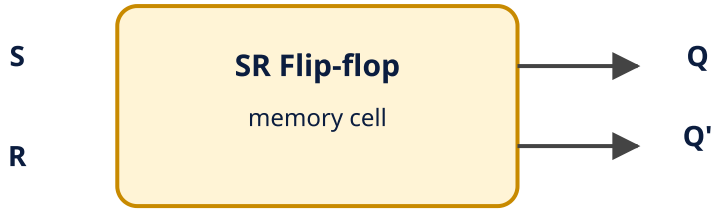
🧮 **Worked example: For A = 1, B = 1, Cin = 1 find S and Cout.**
S = 1 ⊕ 1 ⊕ 1 = **1**. Cout = **1**. ($1+1+1 = 11_2$.)

You can build a full adder from **two half adders plus an OR**: first HA adds A and B → S1, C1; second HA adds S1 and Cin → S, C2; then Cout = C1 + C2.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

4. Storing bits – feedback and flip-flops

A combinational circuit's output depends **only on the current inputs**. To **remember** a bit you need a **feedback loop**: feed the output back to an input so the circuit can hold its state. A **flip-flop** is the basic memory cell that stores **one bit**.



Feedback loop stores one bit

SR flip-flop stores one bit with feedback

The simple **SR (Set-Reset) flip-flop** has inputs S and R and outputs Q and Q′:

- S = 1, R = 0 → **Set**: Q becomes 1.
- S = 0, R = 1 → **Reset**: Q becomes 0.
- S = 0, R = 0 → **Hold**: Q stays as it was (memory!).
- S = 1, R = 1 → **Forbidden** in a basic SR latch (both outputs fight).

💡 Registers, cache and static RAM are banks of flip-flops. Dynamic RAM uses tiny capacitors instead, but the idea – store one bit per cell – is the same.

🧱 **Worked example: A flip-flop currently holds Q = 0. What are S and R to (a) set it to 1, (b) keep it at 0?**
(a) S = 1, R = 0 (set). (b) S = 0, R = 0 (hold) – or actively reset with S = 0, R = 1.

🚀 **Beyond the syllabus (2026 update)** *Not required for the exam – read for interest.* In 2026, a phone CPU still uses billions of CMOS full-adder cells in its ALU and SIMD units, and SRAM caches built from 6-transistor flip-flop-like cells. Emerging MRAM and ReRAM try to replace some DRAM, but the syllabus idea stays: adders compute, flip-flops remember.

Key Words

Half adder – Adds two bits: S = A ⊕ B, C = A · B. No Cin.

Full adder – Adds A, B and Cin: S = A⊕B⊕Cin, Cout = majority.

Sum bit – The result bit of an adder column (XOR of the inputs).

Carry – The extra bit that goes to the next column when the sum ≥ 2.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

Cin / Cout – Carry in from the previous column; carry out to the next.	**Flip-flop** – A memory cell that stores one bit using feedback.
Feedback loop – Wiring an output back to an input so the circuit can hold state.	**SR flip-flop** – Set-Reset latch: S sets Q=1, R resets Q=0, 00 holds.
Set – Make Q = 1 (S = 1, R = 0).	**Reset** – Make Q = 0 (S = 0, R = 1).
Hold / memory – S = R = 0 keeps the previous Q.	**Register** – A group of flip-flops storing a multi-bit word in the CPU.

Sample Q&A

1. Write the Boolean expressions for a half adder.

> **Answer:** S = A ⊕ B, C = A · B.

2. Why does a multi-bit adder need full adders, not only half adders?

> **Answer:** Higher bit columns must accept a carry-in from the previous column; a half adder has no Cin.

3. Write S and Cout for a full adder.

> **Answer:** S = A ⊕ B ⊕ Cin; Cout = A·B + B·Cin + A·Cin.

4. What does a flip-flop store?

> **Answer:** One bit, held by a feedback loop.

5. In an SR flip-flop, what do S = 1, R = 0 and S = R = 0 do?

> **Answer:** S=1,R=0 sets Q to 1; S=R=0 holds the previous value.

6. How can you build a full adder from half adders?

> **Answer:** Two half adders plus an OR gate for the two carry outputs.

MCQ

1. Half-adder Sum equals

A. A ⊕ B ⊕ Cin
B. A · B
C. A ⊕ B
D. A + B
E. A′

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

2. Half-adder Carry equals

A. A · B
B. A + B
C. A ⊕ B ⊕ Cin
D. A′ · B′
E. A ⊕ B

3. A full adder has how many inputs?

A. 1
B. 8
C. 4
D. 2
E. 3 (A, B, Cin)

4. Full-adder Sum equals

A. A · B
B. A ⊕ B ⊕ Cin
C. A ⊕ B
D. A + B
E. majority

5. Full-adder Cout equals

A. A + B + Cin only as OR of three
B. A ⊕ B ⊕ Cin
C. majority of A, B, Cin
D. A′
E. A ⊕ B

6. For A=1,B=1 on a half adder, S and C are

A. S=1, C=1
B. S=0, C=0
C. S=1, C=0
D. S=0, C=1
E. S=1, C=undefined

7. A flip-flop stores

A. no data
B. an analog voltage only
C. one word only
D. one bit
E. one byte

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

8. Feedback in a flip-flop is used to

A. add two numbers
B. hold / remember the state
C. invert a bit only
D. create Cin
E. delete memory

9. SR flip-flop: S=1, R=0 means

A. Hold
B. Toggle randomly
C. Forbidden
D. Reset Q = 0
E. Set Q = 1

10. SR flip-flop: S=R=0 means

A. Hold previous Q
B. Reset
C. Set
D. Q becomes 1 always
E. Forbidden

11. Which is forbidden in a basic SR latch?

A. S = 1, R = 0
B. S = 1 and R = 1
C. S = 0, R = 1
D. Q = 1
E. S = 0, R = 0

12. An 8-bit adder needs how many full adders (roughly)?

A. 1
B. 16
C. 8
D. 4
E. 2

13. Which term means: "Adds two bits: S = A ⊕ B, C = A · B. No Cin"?

A. Sum bit
B. Hold / memory
C. Flip-flop
D. SR flip-flop
E. Half adder

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

14. Which term means: “Adds A, B and Cin: S = A⊕B⊕Cin, Cout = majority”?

A. Full adder
B. Register
C. Set
D. Feedback loop
E. Carry

15. Which term means: “The result bit of an adder column (XOR of the inputs)”?

A. Half adder
B. Reset
C. SR flip-flop
D. Sum bit
E. Cin / Cout

16. Which term means: “The extra bit that goes to the next column when the sum ≥ 2”?

A. Full adder
B. Flip-flop
C. Hold / memory
D. Carry
E. Set

17. Which term means: “Carry in from the previous column; carry out to the next”?

A. Feedback loop
B. Register
C. Sum bit
D. Reset
E. Cin / Cout

18. Which term means: “A memory cell that stores one bit using feedback”?

A. SR flip-flop
B. Flip-flop
C. Hold / memory
D. Half adder
E. Carry

19. Which term means: “Wiring an output back to an input so the circuit can hold state”?

A. Register
B. Full adder
C. Feedback loop
D. Set
E. Cin / Cout

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

20. Which term means: “Set-Reset latch: S sets Q=1, R resets Q=0, 00 holds”?

A. SR flip-flop
B. Half adder
C. Reset
D. Sum bit
E. Flip-flop

Structured Questions

1. Draw the truth table of a half adder and write S and C as Boolean expressions. *(2 marks)*

> Table AB→SC: 00→00, 01→10, 10→10, 11→01 (2); S=A⊕B, C=A·B (2).

2. Explain why a full adder is needed for bits above the LSB. *(3 marks)*

> Those columns receive a carry from the right (1); half adder has no Cin (1); full adder accepts Cin and produces Cout (1).

3. For A=1, B=1, Cin=0 find S and Cout of a full adder. Show working. *(3 marks)*

> S=1⊕1⊕0=0 (1); Cout=AB+BCin+ACin=1+0+0=1 (1); check $1+1+0=10_2$ (1).

4. Describe how a flip-flop stores a bit and what S=R=0 means in an SR flip-flop. *(3 marks)*

> Feedback loop holds state (1); stores one bit (1); S=R=0 is hold / memory of previous Q (1).

5. State the four SR input combinations and their effects. *(4 marks)*

> 10 set Q=1 (1); 01 reset Q=0 (1); 00 hold (1); 11 forbidden (1).

Summary

- Half adder: S = A ⊕ B, C = A · B (no Cin).
- Full adder: S = A ⊕ B ⊕ Cin, Cout = majority(A,B,Cin).
- Chain full adders to add multi-bit numbers in the ALU.
- A flip-flop uses feedback to store one bit.
- SR: 10 set, 01 reset, 00 hold, 11 forbidden.

MCQ answer key – 4.4

1. **C** – XOR of the two bits.
2. **A** – AND of the two bits.
3. **E** – A, B and carry-in.
4. **B** – XOR of all three.
5. **C** – AB + BCin + ACin.
6. **D** – $1+1 = 10_2$.

© 2026 schoolict.net (Ruwan Mangala Suraweera) · All rights reserved

7. **D** – Basic memory cell = 1 bit.
8. **B** – Output fed back keeps Q.
9. **E** – Set input active.
10. **A** – Memory / hold state.
11. **B** – Both set and reset at once.
12. **C** – One per bit column.
13. **E** – Half adder – Adds two bits: S = A ⊕ B, C = A · B. No Cin.
14. **A** – Full adder – Adds A, B and Cin: S = A⊕B⊕Cin, Cout = majority.
15. **D** – Sum bit – The result bit of an adder column (XOR of the inputs).
16. **D** – Carry – The extra bit that goes to the next column when the sum ≥ 2.
17. **E** – Cin / Cout – Carry in from the previous column; carry out to the next.
18. **B** – Flip-flop – A memory cell that stores one bit using feedback.
19. **C** – Feedback loop – Wiring an output back to an input so the circuit can hold state.
20. **A** – SR flip-flop – Set-Reset latch: S sets Q=1, R resets Q=0, 00 holds.

© 2026 schoolict.net (Ruwan Mangala Suraweera). All rights reserved.