Logical and arithmetic shifts

2.1.4 · Data · revision notes

Shifting the bits of a binary number left or right, what this does to its value, and the difference between logical and arithmetic shifts. Expect questions that ask for the result of a shift, its effect on the value, and why a shift can give a wrong or rounded answer.

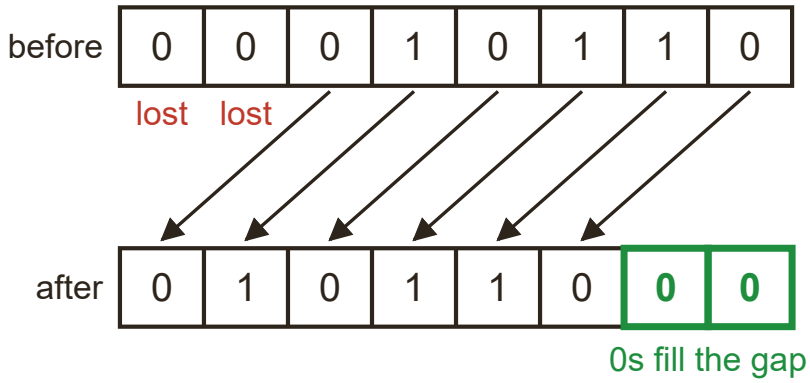
Key facts

- Logical shift left n places: 0s in on the right; value × 2^n
- Arithmetic shift right: copies of the sign bit (MSB) in on the left
- Bits shifted off either end are lost
- × 2 = 1 place, × 4 = 2 places, × 8 = 3 places, × 16 = 4 places
- Logical shift right n places: 0s in on the left; value ÷ 2^n, rounded down
- Arithmetic shifts are for two's complement (signed) numbers
- A 1 lost off the left: overflow (wrong result); a 1 lost off the right: loss of precision

Notes

Logical shifts

- A **binary shift** moves every bit in a binary number a number of places to the left or right. The number still has 8 bits.
- Bits that are shifted off the end are lost.
- **Logical shift left**: the bits move left and 0s fill the empty places on the right. 0001 0110 (22) shifted left 2 places gives 0101 1000 (88).



Logical shift left 2 places: every bit moves 2 places left, the two bits at the left end are lost and 0s fill the gap (22 becomes 88)

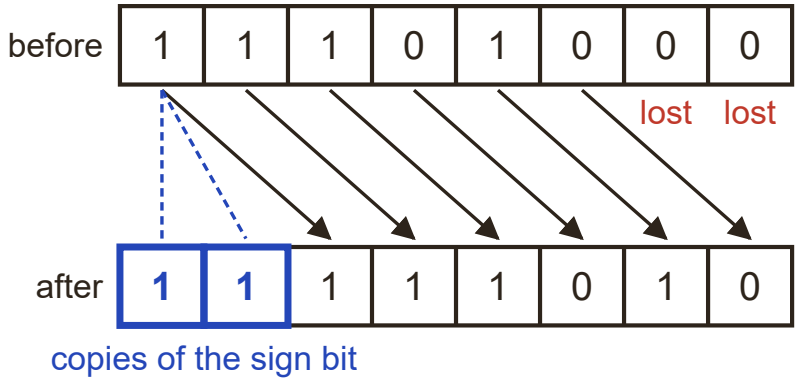
- **Logical shift right**: the bits move right and 0s fill the empty places on the left. 0110 0100 (100) shifted right 2 places gives 0001 1001 (25).

The effect on the value

- Shifting left by n places multiplies an unsigned number by 2^n: 1 place × 2, 2 places × 4, 3 places × 8.
- Shifting right by n places divides by 2^n.
- Left shift problem: if a 1 is shifted off the left, the result is wrong because the true answer does not fit in 8 bits (an overflow). 1010 0000 (160) shifted left 1 place gives 0100 0000 (64), not 320.
- Right shift problem: if a 1 is shifted off the right, precision is lost and the answer is rounded down. 0001 0111 (23) shifted right 1 place gives 0000 1011 (11), not 11.5.

Arithmetic shift right

- An arithmetic shift right is used for two's complement (signed) integers.
- The bits move right, and the empty places on the left are filled with copies of the **sign bit** (the MSB), not with 0s.
- This keeps the sign, so a negative number stays negative and is divided by 2 for each place. 1110 1000 (−24) arithmetic shift right 2 places gives 1111 1010 (−6).



Arithmetic shift right 2 places: the gaps on the left are filled with copies of the sign bit, so −24 becomes −6

- A logical shift right of the same number gives 0011 1010 (+58): the sign is lost, so the answer is wrong.
- For a positive number (MSB 0), logical and arithmetic shifts right give the same result.
- Right shifts always round down, so a negative result is rounded to the more negative whole number: −5 (1111 1011) shifted right 1 place gives 1111 1101, which is −3, not −2.5. Grade 8+ only

How to answer each type of question

Give the result of a shift

1 mark each | **Grade 5**

1. Write out the 8 bits and move each one the stated number of places.
2. Fill the gaps: 0s for a logical shift; copies of the sign bit for an arithmetic shift right.
3. Drop the bits that fall off the end, and check that you still have exactly 8 bits.

Example. The 8-bit binary number 1001 1010 is stored in a register. Give the result of:
(a) a logical shift left by 1 place
(b) a logical shift right by 3 places
(c) an arithmetic shift right by 2 places.

Model answer.

(a) 0011 0100 (1)
(b) 0001 0011 (1)
(c) 1110 0110 (1)

State the effect of a shift on the value

1 to 2 marks | **Grade 5**

1. Count the places and the direction.
2. Left: multiply by 2 for each place. Right: divide by 2 for each place.
3. Give the actual multiplier or divisor, e.g. 'multiplied by 8'.

Example. The 8-bit binary number 0000 1100 represents 12. It is shifted left by 3 places.
(a) Give the result.
(b) State the effect of this shift on the value.

Model answer.

(a) 0110 0000 (1)
(b) It is multiplied by 8 (12 becomes 96) (1)

Explain why the result of a shift is not exact 2 marks Grade 7

1. Work out what the exact answer should be.
2. Give the value the shift actually produces.
3. Say which bits were lost (off the right or off the left) and what that did to the value.

Example. A program divides 8-bit unsigned integers by 4 using a logical shift right by 2 places. It is used on 0010 1011 (43).
Explain why the result is not exactly 43 ÷ 4.

Model answer. 43 ÷ 4 = 10.75, but the shift gives 0000 1010, which is 10 (1). The two 1s shifted off the right-hand end are lost, so the fraction is lost and the answer is rounded down (precision is lost) (1).

Explain the difference between a logical and an arithmetic shift right 4 marks Grade 8

1. Say what fills the gaps in each type of shift.
2. Apply both shifts to the number you are given.
3. Convert both results to denary and say which one is correct and why.

Example. Explain the difference between a logical shift right and an arithmetic shift right.
Use the 8-bit two's complement integer 1101 0000 (−48) to illustrate your answer.

Model answer. A logical shift right fills the empty places on the left with 0s (1). An arithmetic shift right fills them with copies of the sign bit (1). A logical shift right by 1 place gives 0110 1000, which is +104, so the sign has been lost (1). An arithmetic shift right by 1 place gives 1110 1000, which is −24, so the number has been halved correctly and stays negative (1).

Don’t lose marks

- Ending with 7 or 9 bits. A shift never changes the number of bits, so always finish with 8.
- Filling the gaps with 1s in a logical shift. Logical shifts always fill with 0s.
- Filling with 0s in an arithmetic shift right of a negative number. The gaps must copy the sign bit, which is 1.
- Saying a right shift always divides exactly. Bits lost off the right mean the result is rounded down.
- Mixing up the direction: multiplying by 4 is a shift LEFT by 2 places.

More tips

Memory tricks

- Left makes it larger, right reduces it: a left shift multiplies, a right shift divides.
- Places to multiplier: 1 place = 2, 2 places = 4, 3 places = 8. Double once for each place.
- Arithmetic means 'keep the sign': copy the MSB into every gap on the left.
- Check with denary: convert before and after, and see whether the value was multiplied or divided by the right power of 2.

Exam technique

- Write the result directly under the original number so you can see that every bit has moved the right number of places.
- When asked for the 'effect', give the multiplier or divisor (e.g. 'divided by 4'), not just 'it gets smaller'.
- If the number is two's complement and it must be divided, use an arithmetic shift right.
- For 'explain why the answer is wrong', say which bits were lost and what that did to the value.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 4 — **Apply a logical shift left**
Move every bit left by the given number of places and fill the gaps on the right with 0s.

Grade 4 — **Apply a logical shift right**
Move every bit right by the given number of places and fill the gaps on the left with 0s.

Grade 5 — **State the effect of a shift on value**
Each place left multiplies an unsigned number by 2; each place right divides it by 2.

Grade 6 — **Apply an arithmetic shift right**
Move every bit right and fill the gaps on the left with copies of the sign bit (the MSB).

Grade 7 — **Explain loss of precision in a right shift**
Bits shifted off the right-hand end are lost, so an answer that should have a fraction is rounded down.

Grade 7 — **Explain overflow in a left shift**
A 1 shifted off the left-hand end is lost, so the result is wrong because it is too big for 8 bits.

Grade 8 — **Explain why arithmetic shifts suit signed numbers**
Copying the sign bit keeps a negative number negative, so it is correctly divided by 2 for each place.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.