InkPhysics

ANSWER BOOK

Digital signal processing

Electronics · Year 13

AQA 3.13.3

17 questions · 50 marks

NAVY EDITION

inkphysics.co.uk

SECTION A · Warm-up

A1 An analogue signal carries its information in its exact shape, so any noise added on the way becomes part of the signal and cannot afterwards be told apart from it (1). A digital signal has only two meaningful levels, and the receiver asks only whether each pulse is above or below a threshold, so noise smaller than half a logic level leaves the decision unchanged (1). Answering that digital signals are 'sent at higher power' is the standard error, and it is wrong twice over: digital links usually use less power. [2]

A2 An amplifier multiplies whatever arrives, so it boosts the accumulated noise along with the signal and every leg of the journey leaves the signal permanently worse (1). A regenerator decides whether each incoming pulse is a 0 or a 1 and issues a brand-new, full-height, noise-free pulse, discarding the noise collected so far entirely (1). Saying a repeater 'amplifies the digital signal' is the standard error; it replaces it. [2]

A3 A = 0 B = 0 → 0
A = 0 B = 1 → 1
A = 1 B = 0 → 1
A = 1 B = 1 → 0
The output is 1 when the inputs **differ** (1) (1). The row that catches people is A = 1, B = 1, where OR gives 1 but EOR gives 0; confusing the two tables is the standard error. [2]

A4 A = 0 B = 0 → 1
A = 0 B = 1 → 1
A = 1 B = 0 → 1
A = 1 B = 1 → 0
NAND is AND with the output inverted, drawn with a small circle on the output (1) (1). Every other gate can be built from NAND gates alone. Writing the AND table and forgetting to invert it is the standard error, and the give-away is the first row: NAND outputs 1 when both inputs are 0. [2]

A5 On each rising edge of the clock it copies whatever is on the data input D to its output Q (1), and it then holds Q frozen until the next clock edge, ignoring D completely in between (1). That is what makes it a one-bit memory; registers and computer memory are rows of them. Saying the output 'follows the input' is the standard error: it follows D only at the clock edge. [2]

A6 $f_{out} = f_{in}/2^n$ (1). Every stage halves the frequency once. Confusing this with the number of states, which is 2^n rather than a division, is the standard slip. [1]

SECTION B · Standard

B1 [3]

Three inputs give $2^3 = 8$ rows, taken in binary order (1):

A = 0 B = 0 C = 0 → 0
A = 0 B = 0 C = 1 → 0
A = 0 B = 1 C = 0 → 1
A = 0 B = 1 C = 1 → 0
A = 1 B = 0 C = 0 → 1
A = 1 B = 0 C = 1 → 0
A = 1 B = 1 C = 0 → 1
A = 1 B = 1 C = 1 → 0

Q = 1 for 3 of the eight combinations (1) (1). Note that C = 1 forces Q = 0 whatever A and B do, because the NOT C term feeds an AND. Writing only four rows, or listing the inputs in some order of your own, is the standard error: examiners mark the complete table in binary counting order and a missing row forfeits the mark.

B2 [3]

Each stage halves the frequency, so ten stages divide by $2^{10} = 1024$ (1). $f_{out} = 1.024 \times 10^6/1024$ (1) = 1000 Hz (1). Dividing by 10 rather than by 2^{10} is the standard error, and it gives an answer a hundred times too large.

B3 [3]

Division needed = $2.048 \times 10^6/250 = 8192$ (1). $8192 = 2^{13}$ (1), so **13** stages are needed (1). Reaching for a calculator's log button and quoting a non-integer number of stages is the standard error: divisions of this kind are always exact powers of two, and if yours is not, the arithmetic has gone wrong.

B4 [3]

Number of states = $2^6 = 64$ (1), displayed as the counts 0 to 63 (1). Final output frequency = $4096/2^6 = 64$ Hz (1). Quoting the count range as 1 to 64 is the standard error; so is swapping the two answers, since the frequency divides by 2^n while the number of states is 2^n.

B5 [3]

The sum comes from an **EOR** gate, since the sum is 1 exactly when the inputs differ, and the carry from an **AND** gate, since the carry is 1 exactly when both inputs are 1; both share the same pair of inputs (1) (1).

A = 0, B = 0 → sum 0, carry 0
A = 0, B = 1 → sum 1, carry 0
A = 1, B = 0 → sum 1, carry 0
A = 1, B = 1 → sum 0, carry 1

Every row agrees with ordinary arithmetic, the carry read before the sum (1). Naming an OR gate for the sum is the standard error, and it fails on the last row, where 1 + 1 must give sum 0 carry 1.

B6 [3]

Q = D AND R AND NOT K (1), needing one NOT gate feeding a three-input AND, or two two-input ANDs in series (1).
D = 1 R = 1 K = 0 → 1
D = 1 R = 1 K = 1 → 0
D = 0 R = 1 K = 0 → 0
Only the first case sounds the alarm (1). The standard error is to translate 'and the key is not inserted' as an OR, or to drop the NOT and let an inserted key set the alarm off; read the sentence one connective at a time.

B7 [2]

The pulse arrives at 4.6 - 0.9 = 3.7 V (1). That is still above the 2.5 V threshold, so it is read as logic 1 and the pulse is **correct**; the margin available was 2.1 V (1). Judging the pulse against the full 5.0 V rather than against the threshold is the standard error: a digital receiver never asks how close the level is to ideal, only which side of the threshold it lies.

SECTION C · Stretch

C1 [7]

Three inputs, so eight rows in binary counting order (2):
L = 0 S = 0 T = 0 → 0
L = 0 S = 0 T = 1 → 1
L = 0 S = 1 T = 0 → 0
L = 0 S = 1 T = 1 → 1
L = 1 S = 0 T = 0 → 1
L = 1 S = 0 T = 1 → 1
L = 1 S = 1 T = 0 → 0
L = 1 S = 1 T = 1 → 1
The pump runs in 5 of the eight combinations (1). Reading the specification straight off gives P = (L AND NOT S) OR T (2). The circuit needs one NOT gate on S, one AND combining L with NOT S, and one OR bringing in T (1) (1). Check the two authorities in the table: with T = 0 the stop button always wins, and with T = 1 the output is 1 in every row, which is exactly what 'whatever the other two inputs are doing' means. Two standard errors sink this design: feeding T into the AND, which lets the stop button defeat the test, and dropping the NOT so that pressing stop runs the pump.

C2 [4]

4 194 304 = 2^{22} (1), so **22** stages take the frequency down to exactly 1 Hz (1).
After twelve stages the frequency is 4 194 304/2^{12} = 4 194 304/4096 (1) = 1024 Hz (1).
Crystal frequencies are chosen as powers of two for precisely this reason. Dividing by 12 rather than by 2^{12} is the standard error at this stage of the chain.

C3 The first half-adder adds A and B, producing an intermediate sum and an intermediate carry (1). The intermediate sum and the incoming carry go into the second half-adder, whose sum is the final sum bit; the two intermediate carries are combined by the OR gate to give the final carry out (1). For A = 1, B = 1, carry in 1: the first half-adder gives sum 0 and carry 1; the second adds that 0 to the incoming 1, giving sum 1 and carry 0; the OR gate passes the first carry, so carry out = 1 (1). Read together the answer is binary 11, the number three, which is 1 + 1 + 1 (1). Feeding the carry into the **first** half-adder, which has nowhere to take a third input, is the standard error. [4]

C4 Distance before the noise reaches 1.2 V = 1.2/0.35 = 3.43 km, so regenerators go every **3** km (1). A 30 km route is then 30/3 = 10 sections, which needs 9 regenerators along the way, the receiver ending the last section (1). Each regenerator reads every pulse as 0 or 1 while the noise is still well inside the margin and issues a clean new pulse, so the noise never accumulates beyond one section's worth (1). On an analogue link the noise is indistinguishable from the signal, so an amplifier boosts both and the degradation is permanent and cumulative (1). Rounding 3.43 km up to 4 km is the standard error here: the spacing must be rounded **down**, or the noise passes the margin before the next regenerator is reached. Quoting 10 regenerators instead of 9 is the other, since the far end of the last section is the receiver itself. [4]