

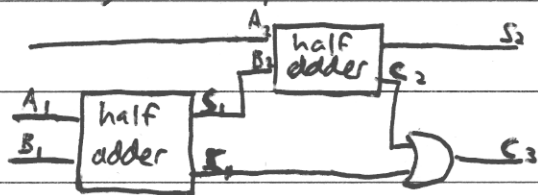


Question 23

a) i)

A	B	A and B C	A or B	Not(A and B)	A and B S
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	1	0	0

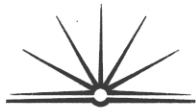
ii) If you connect two half adders up, as in



the diagram you can get

a full adder. A_1 and B_1 are

the number to be added and A_2 is the carry from the last sum (if any). S_1 (the sum of A_1 and B_1) goes into half adder 2 as B_2 . S_2 (the sum of A_2 and B_2) is the ultimate sum of A_1 , B_1 and the carry C_1 and C_2 then go into an OR gate and become C_3 (the end carry). So this is the sum of A_1 , B_1 and the carry to come out as the sum, S_2 , and the carry, C_3 .

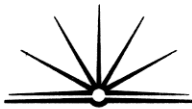


Question 25

b) An integer representation of a number cannot go into decimal points i.e. 2.56 is not an integer because an integer has to be a whole number. A floating point number, however, can go into decimal places i.e. 0.0183 can be represented as an IEEE 32 bit floating point number, 1.83×10^{-2} .

Another difference is that as an integer goes higher it takes up more space i.e. numbers above 128 can no longer be represented in 16 bit form. This is opposed to a floating point number which can represent huge numbers i.e. 6.859×10^{120} .

The floating point number system is best if used with numbers above 128 and decimals as they only take 32-bits of space where as a ^{whole} number under 128 should be represented by an integer as it only takes up 16 bits.



Question 25

83
50
33

c) i) 0101100101

the eight data bits are 10110010 ^{32 16 8 4 2 1} $2+32+16=50$

the car moves right 50mm

0110100111

the eight data bits are 11010011 ^{64 32 16 8 4 2 1} $1+2+16+64=83$

then car moves up 83mm

ii) ~~10110010 + 11010011 = 110000101~~ $13 = 00001101$

~~11010011~~

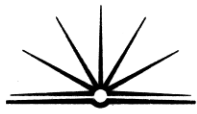
~~= 110000101~~

~~1101
0010
0011~~

~~\therefore checksum = $1101 \overline{) 110000101} \begin{matrix} 000011101 \\ 11000 \\ 0011 \\ 106110 \\ 0011 \\ 10611 \\ 0011 \\ 1100 \\ 0011 \\ 1100 \end{matrix}$~~

\therefore checksum = 1100

← sorry please mark things that are crossed out



Question 25

c)iii) Begin data stream

Create array stream

index = 1

while index \leq 32

Stream(index) = Getbit

index = index + 1

End while

index = 2

while index \leq 9

1st data = 1st data + Stream(index)

index = index + 1

End while

index = 11

while index \leq 19

2nd data = 2nd data + Stream(index)

index = index + 1

End while

index = 26

while index \leq 29

3rd data = 3rd data + stream(index)

Question 25

c) iii) continued.

index = index + 1

End while

If stream(32) = 1 and 3rd data =

(1st data + 2nd data) / 13 then

If 1st data (1) = 1 then

move car 1st data + 128 mm right

Else

move car 1st data mm left

End IF

If 2nd data (1) = 1 then

move car 2nd data + 128 mm up

Else

move car 2nd data mm down

End IF

Else

Flag error

End IF

End data stream.