

HOMEWORK I

DUE : February 9, 2012

READ :

- È Related portions of Chapter 1
- È Related portions of Chapter 2

ASSIGNMENT : There are seven questions four of which are from Chapters I and II of the text-book

Solve all homework and exam problems as shown in class and past exam solutions

1) Solve Problem 1.5.

Can you think of an analog system that may stay as analog for some time to come in the future ?

2) Solve Problem 2.5 (e).

Assume that the binary number is an unsigned binary number. Show manual calculations, indicating that you did not use a calculator.

3) Solve Problem 2.10 (b).

First, convert the hex digits to bit patterns. Second, by assuming that these bit patterns represent 2's complement numbers, perform the addition in binary during which show all the carries. State if there is an overflow and why.

4) Solve Problem 2.18 (f).

The question asks about the radix of the following operation : $\sqrt{(41)_7} = (5)_7$. Both sides of the equation have the **same** radix ! Clearly describe how you arrive at the result.

Hint : Try to make use of the **general conversion formula**.

5) Perform the following operation in 2's complement arithmetic, by using 12 bits per number :

$$(192)_{10} + (F08)_{\text{Hex}} = (?)_{10}$$

You will show all number conversions as done in class. Note that the Hex digits are for coding purposes. They represent a 2's Complement number. Make observations on the addition.

6) Convert the following **fixed-point** decimal number to a 16-bit 2's complement fixed-point binary number : $(525.3125)_{10}$

Use **four** bits for the fraction part of the 2's complement number and **12** bits for the integer part.

7) Calculate the following logarithm : $\log_2[(010000)_2 \text{ 2's Complement}] = (?)_{10}$

RELEVANT QUESTIONS AND ANSWERS

Q1) Convert the following decimal number to a 16-bit 2's complement binary number, by using 7 bits for the integer portion : $(-26.75)_{10}$

A1) First, we have to consider $(+26.75)_{10}$ since we cannot directly convert negative numbers. We know that the integer part is obtained by successive divisions and the fraction part is obtained by successive multiplications :

| | |
|--|---|
| <p style="text-align: center;">The integer part :</p> $\left. \begin{array}{l} 26/2 = 13 \ \& \ 0 \ (\text{lsb}) \\ 13/2 = 6 \ \& \ 1 \\ 6/2 = 3 \ \& \ 0 \\ 3/2 = 1 \ \& \ 1 \\ 1/2 = 0 \ \& \ 1 \ (\text{msb}) \end{array} \right\} (11010)_2$ <p style="text-align: center;">$(+26)_{10} \rightarrow (11010)_2$</p> <p style="text-align: center;">$(+26)_{10}$ using 7 bits requires sign extension : $(0011010)_2$</p> | <p style="text-align: center;">The fraction part :</p> $\left. \begin{array}{l} 0.75 * 2 = 1.5 \rightarrow 1 \\ 0.5 * 2 = 1.0 \rightarrow 1 \end{array} \right\} (.11)_2$ <p style="text-align: center;">$(.75)_{10} \rightarrow (.11)_2$</p> <p style="text-align: center;">$(0.75)_{10}$ using 9 bits requires additional zeros to the right : $(110000000)_2$</p> |
| $(+26.75)_{10} = (0011010.110000000)_2$ $(-26.75)_{10} = \overline{(0011010.110000000)}^2 = (1100101.010000000)_2$ | |

Q2) Hex digits are used to represent two numbers that are 16-bit 2's complement numbers :

$$4AF8 - 1B5E = (?)_{10}$$

Perform the subtraction operation, by converting it to a 16-bit addition operation. Show the result in decimal.

A2) First, we convert the digits to bit strings :

$$\underbrace{0100}_4 \underbrace{1010}_A \underbrace{1111}_F \underbrace{1000}_8 \text{ by using 16 bits}$$

$$\underbrace{0001}_1 \underbrace{1011}_B \underbrace{0101}_5 \underbrace{1110}_E \text{ by using 16 bits}$$

In order to convert the subtraction to an addition, we need to take the 2's complement of the second number :

$$\overline{(0001\ 1011\ 0101\ 1110)}^2 = (1110\ 0100\ 1010\ 0010)$$

$$\begin{array}{r} 0100\ 1010\ 1111\ 1000 \\ + 1110\ 0100\ 1010\ 0010 \\ \hline 1\ 0010\ 1111\ 1001\ 1010 \\ \leftarrow c_{out} \end{array}$$

We calculate the corresponding decimal number : $0010\ 1111\ 1001\ 1010$
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

By numbering the bit positions from right to left, starting at 0, we convert the binary number to a decimal number :

$$\Rightarrow 2^1 + 2^3 + 2^4 + 2^7 + 2^8 + 2^9 + 2^{10} + 2^{11} + 2^{13} = 2 + 8 + 16 + 128 + 256 + 512 + 1024 + 2048 + 8192 = (12186)_{10}$$

Observation : we added a negative number and a positive number, therefore, there **cannot** be any overflow.

Q3) Perform the following operation in 2's complement arithmetic : **F6 + 49 = ?**

The numbers are shown in the Hexadecimal notation. Thus, first convert the numbers to binary, and then add them. Make observations on the addition.

A3) Replace each hexadecimal digit with four bits to convert them to 2's Complement numbers :

$$\begin{array}{r} F6 \longrightarrow \underbrace{1111}_F \underbrace{0110}_6 \\ 49 \longrightarrow \underbrace{0100}_4 \underbrace{1001}_9 \end{array} \longrightarrow \begin{array}{r} 1111\ 0110 \\ + 0100\ 1001 \\ \hline 1\ 0011\ 1111 \\ \leftarrow c_{out} \end{array} \longrightarrow (3F)_{Hex}$$

There is **no** overflow, since the two numbers added have different sign bits : one is negative and the other is positive. Thus, the result cannot exceed the limits for 8-bit 2's complement numbers : $(-128)_{10}$ and $(+127)_{10}$. The c_{out} bit is the carry out from the leftmost bit position and it is 1.

Q4) Consider the following subtraction on 2's Complement numbers represented in **Hex coding** :

$$\begin{array}{r} \text{8 A} \\ - \quad \text{8} \\ \hline ? \end{array}$$

Without using a calculator, perform the 8-bit **2's Complement Binary subtraction**, by converting it to an 8-bit **addition** as shown in class. Make observations on the overflow. Then, convert the result to a decimal number as shown in class.

Note again that both *Hex-coded* numbers above are **2's Complement Binary** numbers and so you will perform the **subtraction** via an addition.

A4) Since the numbers are Hex coded, first we replace the Hex coded digits with bits. Then, before we convert the subtraction to an addition in the 2's Complement Binary system, we perform a **sign extension** on the second number since it is shorter :

$$\begin{array}{r} \text{8 A} \\ - \quad \text{8} \\ \hline ? \end{array} \quad \left. \begin{array}{l} \text{8} \\ \text{1000} \end{array} \right\} \quad \left. \begin{array}{l} \text{A} \\ \text{1010} \\ \text{1000} \\ \text{8} \end{array} \right\} \quad \begin{array}{r} \text{1000 1010} \\ - \quad \text{1000} \\ \hline ? \end{array} \quad \left. \begin{array}{l} \text{1000 1010} \\ - \text{1111 1000} \\ \hline ? \end{array} \right\}$$

$$\begin{array}{r} \text{1000 1010} \\ + \text{0000 0111} \\ \hline \text{0} \leftarrow \text{1001 0010} \end{array}$$

$c_{in} = 1$

$c_{out} = 0$

We added two numbers with opposite signs. There **cannot** be an overflow. That is, the result is correct !

We can convert the result to decimal. Since the result is **negative**, we have to make it positive first :

$$\overline{\text{1001 0010}}^2 = \text{0110 1110}$$

7 6 5 4 3 2 1 0

$$2^6 + 2^5 + 2^3 + 2^2 + 2^1 = 64 + 32 + 8 + 4 + 2 = (110)_{10} \Rightarrow (-110)_{10}$$

Q5) Consider the following **8-bit addition** on two **2's Complement Binary** numbers where the first number is shown in **Hex coding** :

$$\begin{array}{r} (\text{A } 5)_{\text{Hex}} \\ + \quad \text{1110 1110} \\ \hline ? \end{array}$$

Hex coded

Now we have the following addition :

$$\begin{array}{r}
 01010010 \\
 + 11111010 \\
 \hline
 ?
 \end{array}
 \left. \vphantom{\begin{array}{r} 01010010 \\ + 11111010 \\ \hline ? \end{array}} \right\}
 \begin{array}{r}
 \begin{array}{ccccccc}
 \downarrow & \downarrow & \downarrow & & \downarrow & & \downarrow \\
 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
 + & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
 \hline
 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0
 \end{array} \\
 c_{out} \leftarrow 1
 \end{array}$$

Observation : we know that there cannot be any overflow since we added a positive number (the first number) and a negative number (the second number). We convert the result to decimal directly since the result is positive :

$$\begin{array}{cccccccc}
 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0
 \end{array}
 \rightarrow 2^6 + 2^3 + 2^2 \rightarrow 64 + 8 + 4 \rightarrow (76)_{10} \text{ the result}$$

Q7) Perform the following subtraction operation on the two 8-bit 2's complement binary numbers below, by converting it to an 8-bit addition operation. Also, determine the missing bits and show the result in decimal :

$$\begin{array}{r}
 a \\
 - b \\
 \hline
 s
 \end{array}
 \left. \vphantom{\begin{array}{r} a \\ - b \\ \hline s \end{array}} \right\}
 \begin{array}{r}
 10111000 \\
 - 1???0011 \\
 \hline
 ?101????
 \end{array}
 \rightarrow (?)_{10}$$

A7)

$$\begin{array}{r}
 10111000 \\
 - 1???0011 \\
 \hline
 ?101????
 \end{array}
 \left. \vphantom{\begin{array}{r} 10111000 \\ - 1???0011 \\ \hline ?101???? \end{array}} \right\}
 \begin{array}{l}
 \text{We need to take the 2's complement of (b)} \\
 \text{to convert the subtraction to an addition first :} \\
 \hline
 (b) = \overline{\overline{(1???0011)}} = 0???1101 = (-b)
 \end{array}$$

$$\begin{array}{r}
 (-b) \rightarrow 10111000 \\
 + 00111101 \\
 \hline
 0 \leftarrow 11010101 \\
 c_{out}
 \end{array}
 \left. \vphantom{\begin{array}{r} 10111000 \\ + 00111101 \\ \hline 11010101 \\ c_{out} \end{array}} \right\}$$

The sum is a negative number.
 We cannot convert it to decimal directly.
 We will convert the negative of the sum to decimal :

$$(-sum) = \overline{\overline{(11010101)}} = 00101011 = 2^0 + 2^1 + 2^3 + 2^5 = 1 + 2 + 8 + 32 = (+43)_{10}. \text{ The sum is } (-43)_{10}$$

Number b is also a negative number. We will convert its negative to decimal first :

$$(-b) = 00011101 \Rightarrow 2^0 + 2^2 + 2^3 + 2^4 = 1 + 4 + 8 + 16 = (+29)_{10} \Rightarrow b = (-29)_{10}$$

Note that we added a negative number (a) and a positive number (-b), therefore, there **cannot** be any overflow.

Q8) Without using a calculator, perform the following 8-bit **2's Complement Binary subtraction**, by converting it to an 8-bit **addition** as shown in class :

$$\begin{array}{r} 1000\ 1000 \\ - \quad 10\ 1101 \\ \hline ? \end{array}$$

Make observations on the overflow. Then, convert the result to a decimal number and also *code* the result in **Hexadecimal** as shown in class.

Note again that both numbers above are **2's Complement Binary** numbers and so you will perform the **subtraction** via an addition.

A8) Before we convert the subtraction to an addition in the 2's Complement Binary system, we perform a **sign extension** on the second number since it is shorter and both numbers are 2's Complement numbers :

$$\begin{array}{r} 1000\ 1000 \\ - \quad 10\ 1101 \\ \hline ? \end{array} \quad \begin{array}{r} 1000\ 1000 \\ - 1110\ 1101 \\ \hline ? \end{array} \quad \begin{array}{r} 1000\ 1000 \\ + 0001\ 0010 \\ \hline 1001\ 1011 \\ \leftarrow c_{out} \quad \leftarrow c_{in} \end{array}$$

We added two numbers with opposite signs. There **cannot** be an overflow. That is, the result is correct !

We can convert the result to decimal. Since the result is **negative**, we have to make it positive first :

$$\overline{1001\ 1011}^2 = \begin{array}{r} 0110\ 0101 \\ \hline 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \end{array} \left. \vphantom{\overline{1001\ 1011}^2} \right\} 2^6 + 2^5 + 2^2 + 2^0 = 64 + 32 + 4 + 1 = (101)_{10} \Rightarrow (-101)_{10}$$

The result of the addition in terms of Hex digits : $\left. \begin{array}{r} 1001 \\ \hline 9 \end{array} \quad \begin{array}{r} 1011 \\ \hline B \end{array} \right\} (9B)_{Hex}$

Q9) Consider the following **8-bit subtraction** on two **2's Complement Binary** numbers :

$$\begin{array}{r} 1010\ 1111 \\ - \quad 0010\ 0001 \\ \hline ? \end{array}$$

Without using a calculator, perform the 8-bit **2's Complement Binary subtraction** by converting it to an 8-bit addition as shown in class. Make observations on the overflow. Then, convert the result to a decimal number as shown in class.

A9) We convert the 8-bit subtractions to an addition, by complementing the second number and inputting a 1 via c_{in} . Then, we perform the addition :

$$\begin{array}{r}
 1010\ 1111 \\
 - 0010\ 0001 \\
 \hline
 ?
 \end{array}
 \quad
 \left.
 \begin{array}{r}
 1010\ 1111 \\
 + 1101\ 1110 \\
 \hline
 1\ 1000\ 1110 \\
 \leftarrow c_{out}
 \end{array}
 \right\}
 \begin{array}{l}
 \leftarrow c_{in} \\
 \leftarrow c_{out}
 \end{array}$$

We added two negative numbers and the result is negative. Therefore, there is **no** overflow. That is, the result is correct !
 We can convert the result to decimal. Since the result is **negative**, we have to make it positive first :

$$\overline{1000\ 1110}^2 = \begin{array}{r} 0111\ 0010 \\ 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \end{array} \left. \right\} 2^6 + 2^5 + 2^4 + 2^1 = 64 + 32 + 16 + 2 = (114)_{10} \Rightarrow (-114)_{10}$$

Q10) Consider the following **fixed-point** addition on two **Unsigned Binary** numbers :

$$\begin{array}{r}
 1011.01 \\
 + 0011.10 \\
 \hline
 ?
 \end{array}$$

Remember :
 These numbers are
fixed-point numbers !

Without using a calculator, perform the **fixed-point Unsigned Binary** addition as shown in class. Make observations on the overflow. Then, convert the result to a **decimal** number and also code the result in **Hexadecimal** as shown in class.

A10) We perform the fixed-point unsigned binary addition below :

$$\begin{array}{r}
 1011.01 \\
 + 0011.10 \\
 \hline
 0\ 1110.11 \\
 \leftarrow c_{out}
 \end{array}
 \quad
 \left.
 \begin{array}{l}
 \leftarrow c_{in} \\
 \leftarrow c_{out}
 \end{array}
 \right\}$$

The c_{out} bit is 0. Therefore, there is **no** overflow.
 That is, the result is correct !

We can convert the result to decimal :

$$\begin{array}{r}
 1110.11 \\
 3\ 2\ 1\ 0\ -1\ -2
 \end{array}
 \quad
 2^3 + 2^2 + 2^1 + 2^{-1} + 2^{-2} = 8 + 4 + 2 + 0.5 + 0.25 = (14.75)_{10}$$

We code the result in Hexadecimal :

Since there are two fraction bits, we attach two zeros so that there are four fraction bits

$$\left. \begin{array}{l} 1110 \\ E \end{array} \right\} \left. \begin{array}{l} .1100 \\ C \end{array} \right\} \Rightarrow (E.C)_{Hex}$$

Q11) Calculate the minimum number of bits necessary to represent the following decimal number in the 2's complement system : $(92)_{10}$

A11) First, as we discussed in class we have to check if the number is 0 or -2^x . It is neither. Therefore, we use the formula discussed in class to determine the minimum number of bits needed to represent the decimal number in the **unsigned binary** system :

$$\lceil \log_2(92 + 1) \rceil = \lceil \log_2 93 \rceil = \lceil 6.53 \rceil = 7$$

In the 2's complement system, one additional bit is needed as the sign bit. Therefore, we need at least **8** bits to represent $(92)_{10}$ in the 2's Complement system.

Q12) Determine the base (radix) of the numbers used in the following addition : $13 + 6 = 21$

A12) There are several ways to solve the problem. One that uses the general conversion formula is as follows :

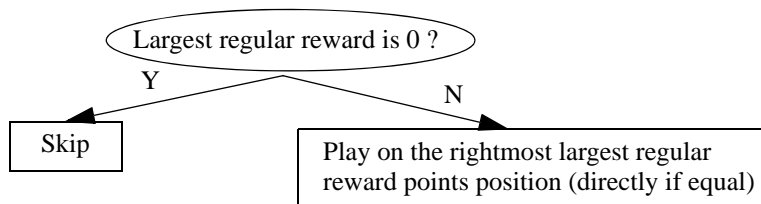
È First we have to see if "6" in the unknown radix is equal to "6" in decimal. By using the general conversion formula we see that :

$$(6)_r = (6)_r = 6 * r^0 = 6 * 1 = (6)_{10}$$

È Then, we do the addition by using the general conversion formula to obtain the unknown radix "r" :

$$\left. \begin{array}{r} 13 \\ + 6 \\ \hline 21 \end{array} \right\} \begin{array}{l} (13)_r + (6)_r = (21)_r \Rightarrow (1 * r^1) + (3 * r^0) + 6 = (2 * r^1) + (1 * r^0) \\ \Rightarrow r + 3 + 6 = 2r + 1 \Rightarrow r + 9 = 2r + 1 \Rightarrow \mathbf{r = 8} \end{array}$$

Q13) Consider the **Ppm** term project. The flowchart of the playing strategy of a machine player is below :



Assume that the code is 6A.

Consider the table below that shows the random digit, position displays **before** and **after** the **machine** player plays, whether the random digit is played directly or added, the number of adjacencies, the points earned by the **machine** player and whether the machine player plays again.

| RD | Displays Before Play PD3 PD2 PD1 PD0 | Displays After Play PD3 PD2 PD1 PD0 | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again |
|----|---|--|-----|---------------|-------------------------|----------------------------|
| 5 | F A A F | F A (F) F | A | 1 | 30 | Yes |
| 2 | C A C C | | | | | |
| 6 | 0 0 0 0 | | | | | |
| 1 | E E F E | | | | | |
| 9 | A 1 A 1 | | | | | |

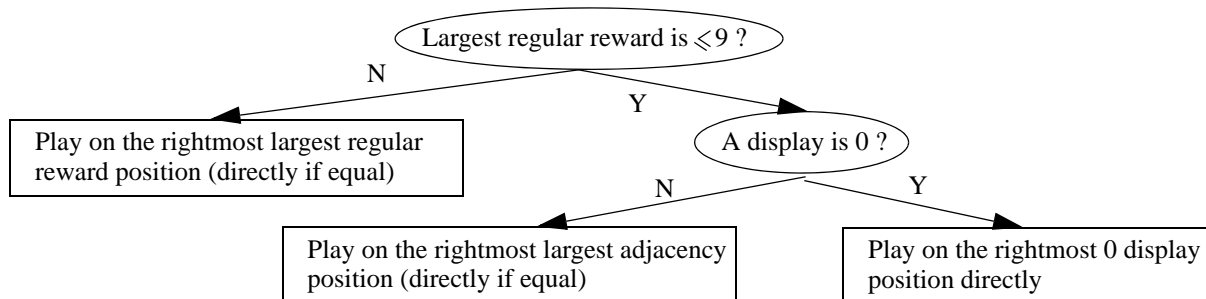
The first row shows how the random digit is played by the **machine** player. A circle is drawn on a position if it is played on. The meaning of **D/A** is Direct/Add which is whether the player plays the random digit **directly** on a position or by **adding** to a position. Note that the cases are **independent** of each other. That is, they do not necessarily follow each other with respect to time. **Continue with the remaining rows.**

A13)

| RD | Displays Before Play PD3 PD2 PD1 PD0 | Displays After Play PD3 PD2 PD1 PD0 | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again |
|----|---|--|-----|---------------|-------------------------|----------------------------|
| 5 | F A A F | F A (F) F | A | 1 | 30 | Yes |
| 2 | C A C C | C (C) C C | A | 3 | 96 | Yes |
| 6 | 0 0 0 0 | 0 0 0 (6) | D | 0 | 6 | No |
| 1 | E E F E | E E F (F) | A | 1 | 30 | Yes |
| 9 | A 1 A 1 | A (A) A 1 | A | 2 | 40 | Yes |

Note that the machine player does not check for code digits and so misses large reward points on rows 3 and 5. The random digits on these rows enable it to play the code digits.

Q14) Consider the **Ppm** term project. The flowchart of the playing strategy of a machine player is below :



Assume that the code is **A3**.

Consider the table below that shows the random digit, position displays **before** and **after** the **machine** player plays, whether the random digit is played directly or added, the number of adjacencies, the points earned by the **machine** player and whether the machine player plays again.

| RD | Displays Before Play PD3 PD2 PD1 PD0 | | | | Displays After Play PD3 PD2 PD1 PD0 | | | | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again |
|----|---|---|---|---|--|---|---|---|-----|---------------|-------------------------|----------------------------|
| 4 | C | 8 | C | C | C | Ⓢ | C | C | A | 3 | 96 | Yes |
| 7 | 1 | 0 | 0 | 1 | | | | | | | | |
| 3 | F | F | 3 | F | | | | | | | | |
| 9 | 6 | 0 | 1 | 3 | | | | | | | | |
| 8 | 2 | A | 2 | 5 | | | | | | | | |
| 6 | 1 | 7 | 7 | 1 | | | | | | | | |

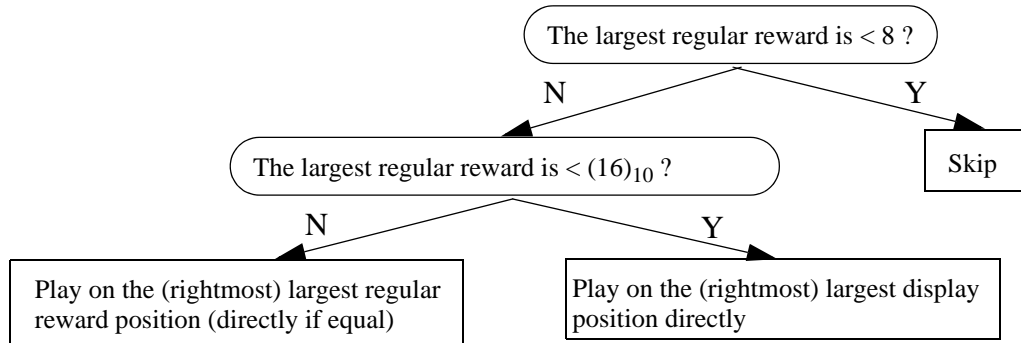
The first row shows how the random digit is played by the **machine** player. A circle is drawn on a position if it is played on. The meaning of **D/A** is Direct/Add which is whether the player plays the random digit **directly** on a position or by **adding** to a position. Note that the cases are **independent** of each other. That is, they do not necessarily follow each other with respect to time. **Continue with the remaining rows.**

A14)

| RD | Displays Before Play PD3 PD2 PD1 PD0 | | | | Displays After Play PD3 PD2 PD1 PD0 | | | | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again |
|----|---|---|---|---|--|---|---|---|-----|---------------|-------------------------|----------------------------|
| 4 | C | 8 | C | C | C | Ⓢ | C | C | A | 3 | 96 | Yes |
| 7 | 1 | 0 | 0 | 1 | 1 | 0 | Ⓢ | 1 | D | 0 | 7 | No |
| 3 | F | F | 3 | F | F | F | 3 | Ⓢ | D | 1 | 30 | Yes |
| 9 | 6 | 0 | 1 | 3 | Ⓢ | 0 | 1 | 3 | A | 0 | 15 | No |
| 8 | 2 | A | 2 | 5 | 2 | A | Ⓢ | 5 | A | 1 | 100 | Yes |
| 6 | 1 | 7 | 7 | 1 | 1 | 7 | 7 | Ⓢ | A | 2 | 28 | Yes |

The machine player does not check for code digits and misses a large reward point on row 4. But, the machine player earns a large amount of reward points on rows 3 and 5 by playing code digits accidentally.

Q15) Consider the **Ppm** term project. The graph of the playing strategy of an imaginary machine player is as follows :



Consider the following table that shows the random digit, position displays **before** and **after** the **machine** player plays, whether the random digit is played directly or added, the number of adjacencies, the points earned by the **machine** player and whether the machine player plays again :

| RD | Displays Before Play PD3 PD2 PD1 PD0 | Displays After Play PD3 PD2 PD1 PD0 | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again ? |
|----|---|--|-----|---------------|-------------------------|------------------------------|
| 1 | F E D E | | | | | |
| 5 | 5 F 6 7 | | | | | |
| 6 | 1 1 1 1 | | | | | |
| 8 | 2 2 2 2 | | | | | |
| 2 | 2 4 4 6 | | | | | |

Assume that the code is **E8**.

The meaning of **D/A** is Direct/Add which is whether the machine player plays the random digit **directly** on a position or by **adding** to a position. A **circle** is drawn on a position if it is played on. Note that the cases are **independent** of each other. That is, they do not necessarily follow each other with respect to time. **Work on the rows.**

A15)

| RD | Displays Before Play PD3 PD2 PD1 PD0 | Displays After Play PD3 PD2 PD1 PD0 | D/A | The Adjacency | Points Earned (Decimal) | Machine player plays again ? |
|----|---|--|------|---------------|-------------------------|------------------------------|
| 1 | F E D E | F E (E) E | A | 2 | 168 | Y |
| 5 | 5 F 6 7 | 5 (5) 6 7 | D | 1 | 10 | Y |
| 6 | 1 1 1 1 | 1 1 1 1 | Skip | Skip | Skip | Skip |
| 8 | 2 2 2 2 | 2 2 2 (8) | D | 0 | 72 | N |
| 2 | 2 4 4 6 | (4) 4 4 6 | A | 2 | 16 | Y |

The machine player strategy does **not** check for code digits and so misses to earn code reward points when the random digit is 2 above. On the other hand, by chance it earns code reward points when the random digit is 1 and 8.