Problema 1

Apartat A El codi ha de ser autocomplementari, i si tots els pesos son positius la codificació 1111 ha de correspondre al valor 9. Per a què això sigui possible, $1 + 2 + 1 + x = 9 \Leftrightarrow x = 5$.

Per tant x, en cas d'existir i ser positiu, ha de ser 5.

Apartat B Assignem codificacions per als primers 5 valors:

$$0000 \to 0$$
$$0010 \to 1$$
$$0100 \to 2$$
$$0110 \to 3$$
$$1110 \to 4$$

La resta de codificacions queden determinades perquè el codi és autocomplementari:

$$0001 \to 5$$
$$1001 \to 6$$
$$1011 \to 7$$
$$1101 \to 8$$
$$1111 \to 9$$

Tot quadra, tenim un codi BCD ponderat autocomplementari vàlid.

Apartat C Una opció és assignar pes -9 al cinquè bit. Llavors, quan estigui activat, les combinacions dels bits existents recorren el rang $[-9, 0]$. Només caldria descartar una de les dues representacions de 0 resultants. El codi podria quedar així:

$$10000 \to -9$$
$$10010 \to -8$$
$$10100 \to -7$$
$$10110 \to -6$$
$$11110 \to -5$$
$$10001 \to -4$$
$$11001 \to -3$$
$$11011 \to -2$$
$$11101 \to -1$$
$$00000 \to 0$$
$$00010 \to +1$$
$$00100 \to +2$$
$$00110 \to +3$$
$$01110 \to +4$$
$$00001 \to +5$$
$$01001 \to +6$$
$$01011 \to +7$$
$$01101 \to +8$$
$$01111 \to +9$$

Que a més té la particularitat de que segueix sent autocomplementari excepte per la doble representació del 0.

Problema 2

Apartat A Escrivim la taula de veritat:

$b_3b_2b_1b_0$	y_5	y_4	y_3	y_2	y_1	y_0
0000	0	0	0	0	0	0
0001	0	0	0	1	0	1
0010	0	0	1	0	1	0
0011	0	0	1	1	1	1
0100	0	1	0	1	0	0
0101	0	1	1	0	0	1
0110	0	1	1	1	1	0
0111	1	0	0	0	1	1
1000	1	0	1	0	0	0
1001	1	0	1	1	0	1
1010	–	–	–	–	–	–
1011	–	–	–	–	–	–
1100	–	–	–	–	–	–
1101	–	–	–	–	–	–
1110	–	–	–	–	–	–
1111	–	–	–	–	–	–

Per a y_0 i y_1, es veu a simple vista que:

$$y_0 = b_0$$
$$y_1 = b_1$$

Per a les altres quatre sortides, fem el mapa de Karnaugh.

Per a y_2:

b_1b_0 / b_3b_2	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	–	–	–	–
10	0	1	–	–

$$y_2 = b_2\overline{b_0} + \overline{b_2}b_0$$

Per a y_3:

b_1b_0 / b_3b_2	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	–	–	–	–
10	1	1	–	–

$$y_3 = b_2\overline{b_1}b_0 + b_1\overline{b_2} + b_1\overline{b_0} + b_3$$

Per a y_4:

b_1b_0 / b_3b_2	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	–	–	–	–
10	0	0	–	–

$$y_4 = b_2\overline{b_0} + b_2\overline{b_1}$$

Per a y_5:

b_1b_0 / b_3b_2	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	–	–	–	–
10	1	1	–	–

$$y_5 = b_3 + b_2b_1b_0$$

Apartat B La funció que estem realitzant té quatre entrades i sis sortides. Cap sortida és suma de més de 4 productes.

Per tant, la PAL mínima per a realitzar el bloc B1 consta de 4 entrades, 4 productes i 6 sortides.

Apartat C A simple vista es veu que una combinació $a_3 ... a_0$ no pertany al codi BCD si $a_3 \cdot (a_2 + a_1) = 1$.

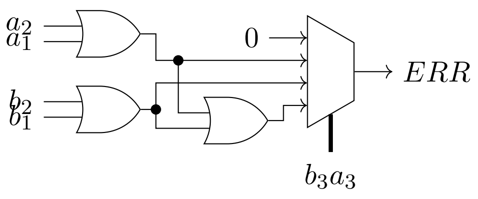
Per tant, el bloc ERR es podria descriure per l'expressió:

$$ERR = a_3 \cdot (a_2 + a_1) + b_3 \cdot (b_2 + b_1)$$

La taula de veritat (parcial) en funció de a_3 i b_3 és:

b_3	a_3	ERR
0	0	0
0	1	$a_2 + a_1$
1	0	$b_2 + b_1$
1	1	$(a_2 + a_1) + (b_2 + b_1)$

El logigrama amb un MUX 4:1 quedaria així:



Apartat D Hem de tenir en compte que desplaçar (emplenant amb zeros) un nombre codificat en binari natural n bits cap a la dreta equival a multiplicar per 2^n el valor codificat. Llavors, el bloc B1 es pot descriure com:

$$Y = \left(B \cdot 2^2\right) + B = 5B$$

La suma no sortirà mai de rang per a cap de les entrades possibles, per tant l'expressió descriu perfectament el comportament del bloc.

Anàlogament, per a B2 tenim:

$$Z = \left(Y \cdot 2^1\right) + A = 2Y + A$$

Per tant, els blocs B1 i B2 conjuntament efectuen l'operació:

$$Z = 10B + A$$

En altres paraules: es retorna a Z el valor, en binari natural, que representen $b_3 ... b_0\, a_3 ... a_0$ codificats en BCD.

Estem davant un *convertidor de BCD natural de dues xifres a binari natural*.

Apartat E Al primer buit:

```
signal P : std_logic_vector(7 downto 0);
```

Al segon buit:

```
P <= "1010" * B;
Z <= P(6 downto 0) + ("000" & A);
```