

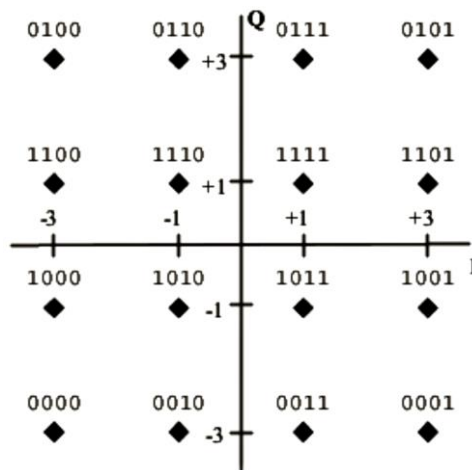
Projet d'application N°3

Chargement des porteuses d'un modulateur DMT

Objet du compte rendu :

L'objectif est d'implémenter une modulation 16QAM.

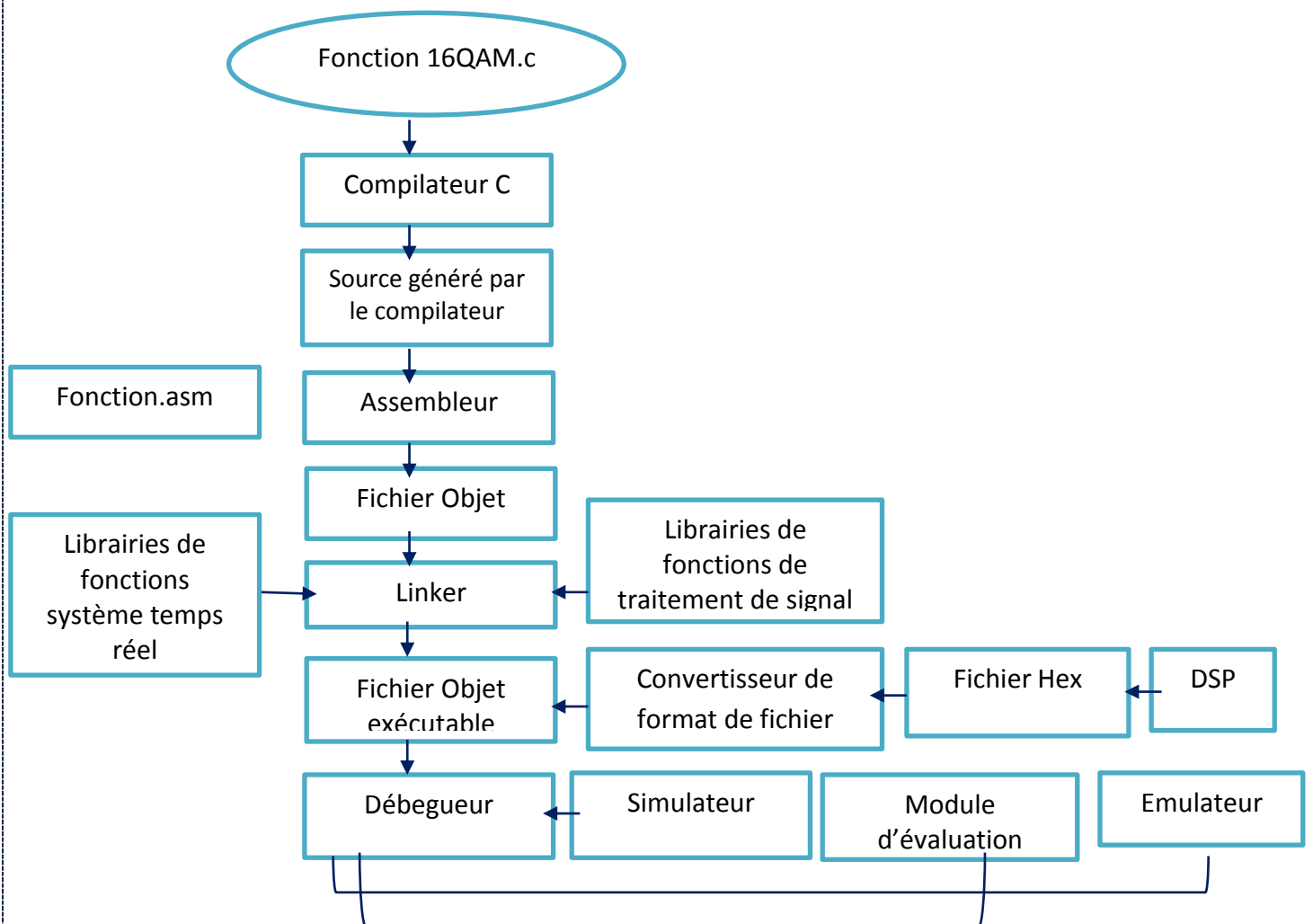
Constellation 16-QAM



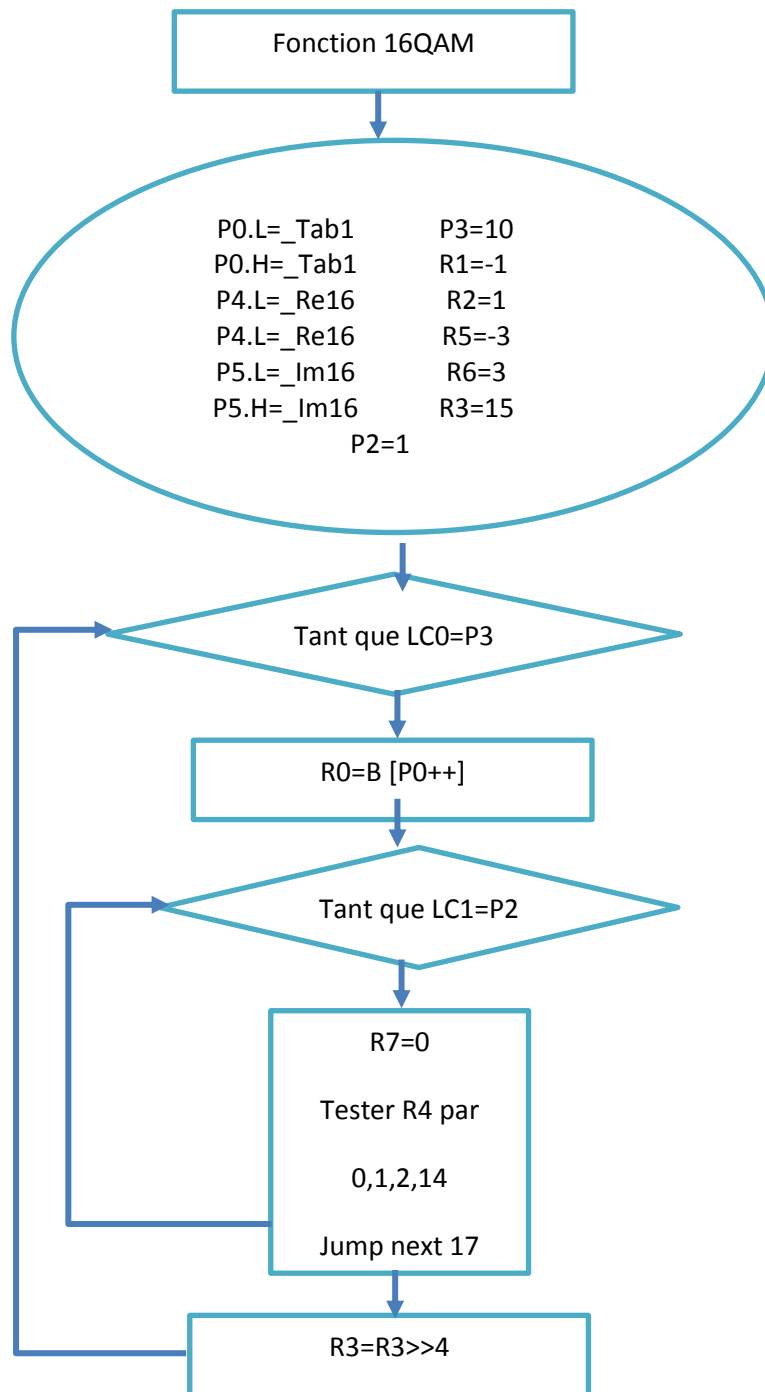
Les deux tableaux Re4 et Im4 sont données par :

bit	Partie réelle	Partie imaginaire
0000	-3	-3
0001	3	-3
0010	-1	-3
0011	1	-3
0100	-3	3
0101	3	3
0110	-1	3
0111	1	3
1000	-3	-1
1001	3	-1
1010	-1	-1
1011	1	-1
1100	-3	1
1101	3	1
1110	-1	1
1111	1	1

1.1 Organigramme d'un système de traitement de données pour DSP :



1.2 Flot de données :



2 Application programmé en assembleur

▪ 16QAM.c

```
#include<stdio.h>
extern int 16QAM();

① int main( )
{
    16QAM();
}
```

▪ 16QAM.asm

```
/****** 16QAM.asm *****/
.section L1_data;
.byte _Tab1[10]={0xAC,0xBA,0xCA,0xCC,0xFD,
0x78,0xA0,0x9A,0x4E,0xE4};
.byte _Re16[20];
.byte _Im16[20];
.global _T16QAM;
.section L1_code;
_16QAM:
nop;
P0.L=_Tab1;
P0.H=_Tab1;
P2=1;
P3=10;
P4.L=_Re16;
P4.H=_Re16;
P5.L=_Im16;
P5.H=_Im16;
R1=-1;
R2=1;
R5=-3;
R6=3;
R3=15;
LSETUP (d2, f2) LC0=P3;
➔ R2: R0 = b[P0++];
R7=0;
LSETUP (d3, f3) LC1=P2;
d3:

R4= R3 & R0;
cc = R4 == R7;
R7=R7+R2;
if !cc jump next3;
b[P4++]=R5;
b[P5++]=R5;
jump f3;

next3:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next4;
b[P4++]=R6;
b[P5++]=R5;
jump f3;
```

```

next4:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next5;
b[P4++]=R1;
b[P5++]=R5;
jump f3;

next5:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next6;
b[P4++]=R2;
b[P5++]=R5;
jump f3;

next6:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next7;
b[P4++]=R5;
b[P5++]=R6;
jump f3;

next7:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next8;
b[P4++]=R6;
b[P5++]=R6;
jump f3;

next8:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next9;
b[P4++]=R1;
b[P5++]=R6;
jump f3;

next9:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next10;
b[P4++]=R2;
b[P5++]=R6;
jump f3;

next10:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next11;
b[P4++]=R5;
b[P5++]=R1;
jump f3;

next11:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next12;
b[P4++]=R6;
b[P5++]=R1;
jump f3;

next12:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next13;
b[P4++]=R1;
b[P5++]=R1;
jump f3;

next13:
cc = R4 == R7;
R7=R7+R2;
if !cc jump next14;
b[P4++]=R2;
b[P5++]=R1;
jump f3;

next14:

```

```

cc = R4 == R7;
R7=R7+R2;
if !cc jump next15;
b[P4++]=R5;
b[P5++]=R2;
jump f3;

next15:

cc = R4 == R7;
R7=R7+R2;
if !cc jump next16;
b[P4++]=R6;
b[P5++]=R2;
jump f3;

next16:

cc = R4 == R7;
R7=R7+R2;
if !cc jump next17;
b[P4++]=R1;
b[P5++]=R2;
jump f3;

next17:

b[P4++]=R2;
b[P5++]=R2;
f3: R3 >>= 4;
f2: nop;
RTS;
_ 16QAM.end:|

```

■ Résultats

Re16																			
IN Memory Sign	_Re16																		
	[FF90000A]	-3	-1	-1	-3	+3	-3	-3	-1	-1	-3	+0	+0	+0	+0	+0	+0	+0	
	[FF90001D]	+0																	
	_Im16																		
	[FF90001E]	+1	-1	-1	+1	+1	-1	-3	-1	+1	+3	+0	+0	+0	+0	+0	+0	+0	
	[FF900031]	+0	+0	+0															
		cpib_ctrl																	

■ Résultats

⇒ Tableau initial

```

Tabl
Tabl
[FF900000] 10100001 00111011 11000011 10011000 00110101 11010011 01010100 00110010 10101100 10110101

```

⇒ Les tableaux réels et imaginaires après modulation

```

Tabl
Tabl
[FF900000] -95 +59 -61 -104 +53 -45 +84 +50 -84 -75
Tabreal
[FF90000A] +1 -1 -1 -1 +1 -1 +1 -1 +1 -1 -1 +1 -1 -1 +1 -1 +1 -1 +1 -1 +1 +1 +1
[FF900026] -1 -1 +1 -1 -1 +1 -1 -1 +1 +1 +1 -1
Tabimag
[FF900032] -1 -1 +1 +1 +1 +1 +1 -1 +1 -1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1 -1 -1
[FF90004E] +1 -1 +1 -1 -1 +1 +1 +1 -1 -1 +1 +1 +0 +0

```