# 

# Constelación 64 QAM

Tabla 1 Normalización de constelaciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Factor Escalado** | **BPSK (tps)** | **BPSK (pilotos)** | **64QAM (datos)** |
|  |  |  |  |  |
|  |  |  |  |  |
| **Símbolo sin normalizar** | 1 | 6.48 | 8.64 | 7 + 7i |
| **Normalizado E(c, c\*) = 16/9** | 8.64 | 0.75 | 1 | 0.81 + 0.81i |

#### 

% Reading the values from Figure 9a

dvbt\_qam = [ +7 + 7i % ++

% Normalization factor from Section 4.4

M = 64; %

v = log2(M); % Constalletion Size

arg = 2/3 \* (M - 1); %

scale = 4/3 \* sqrt(arg); % Scaling factor

dvbt\_qam = dvbt\_qam/scale;

for i=1:M

qam\_re(i) = real(dvbt\_qam(i));

qam\_im(i) = imag(dvbt\_qam(i));

end;

# QAM Mapper (1.0.15)

* El mapeador se implementa mediante dos memorias ROM de 64 x 16 bits.
* Su contenido esta codificado en formato 1.0.15 con lo que el máximo numero representable es ±1.

|  |  |
| --- | --- |
|  | Genera el contenido de las ROM (ETSI 300744 - Figure 9a).  ..\FPGA\_Transmisor\data\MapeadorQAM |
|  | Fichero resultado con el contenido de la ROM real.  ..\ FPGA\_Transmisor \data\MapeadorQAM\release |

% A/D Conversion 1.0.15

dac\_bit= 16;

lsb = 1 / 2^(dac\_bit-1);

for i=1:M

aux\_real(i) = floor(( qam\_re(i) + lsb/2 )/lsb );

aux\_imag(i) = floor(( qam\_im(i) + lsb/2 )/lsb );

end;

diag\_qam64\_adc = aux\_real(:) + j \* aux\_imag(:);