



MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

SISTEMAS ELECTRÓNICOS DE PROCESSAMENTO DE SINAL

BPSK Modem

Grupo n.º 2/3

André Filipe Barroso Cerqueira	n.º 65144
Guilherme Branco Teixeira	n.º 70214
João André Catarino Pereira	n.º 73527

segunda-feira 15h30 - 18h30, LE1

Lisboa, 17 de Abril de 2015

1 Índice

2 Introdução

Este trabalho consiste na primeira parte do projecto de laboratório da cadeira: desenvolvimento de um modem de *Binary Phase Shift Keying* (BPSK).

Teve como objectivo a familiarização com o ambiente de desenvolvimento integrado de DSP que consiste nas placas de desenvolvimento DSK TMS320C6416 e DSK TMS320C6713 da *Texas Instruments* e no software de desenvolvimento *CodeComposerStudio*5.5. Para tal correram-se dois projectos exemplo (`sine8_buf` e `loop_intr`) e, usando as ferramentas de debug, alteraram-se certos parâmetros de forma a observar os efeitos nos sinais resultantes. Também se consolidaram os conhecimentos adquiridos desenvolvendo dois mini projectos: um oscilador sinusoidal controlado numericamente e o modulador do modem BPSK.

3 Projecto

3.1 Projectos de Demonstração

- Resumo das funções e os seus objectivos com especial importancia ao loop
- Relaciona-las com as suas utilizações no projecto em si

3.1.1 `sine8_buf`

O objectivo deste projeto é representar a função sinusoidal, multiplicada por um determinado ganho, através de um conjunto de amostras que equivalem a um período da mesma, repetindo nos períodos seguintes esse mesmo conjunto. Este procedimento é realizado através da rotina de interrupção presente no programa.

Ao analisar o código deste projeto à primeira vista podemos concluir logo que este usa uma frequência de amostragem de 8 kHz , tem um ganho $G = 10$ predefinido e usa 8 amostras para representar a senoide. Depois de observar a senoide no osciloscópio variou-se o ganho a fim de perceber a sua influência e também o seu limite.

Para compreender o limite desta senoide é necessário ter em conta que se usa o formato de vírgula fixa Q15 para as amostras da senoide. Este formato tem como limite o valor $(2^{15} - 1) = 32767$. Considerando o valor máximo da senoide, se multiplicarmos a mesma por um ganho $G=33$ obtemos um valor superior ao permitido pelo formato Q15, fazendo com que nesses pontos o valor da senoide "caia".

3.1.2 `loop_intr`

Este projeto tem como objectivo fornecer-nos um template para os próximos projetos, em termos de comunicação com a placa e rotina de interrupção. Pode-se observar nas últimas linhas de código

como se liga os sinais de entrada e saída aos canais da placa.

(comentario André)No projecto anterior observou-se os efeitos de overflow de uma variável. Neste observam-se os efeitos de aliasing(ou não, não me recordo se o DSP tem um filtro anti-aliasing à entrada) devido ao sinal de line-in ter a mesma frequência que a frequência de amostragem. Se havia anti-aliasing, a partir dos 4khz deixariamos de ver uma senoide com os 3.3V. Não fizemos a experiência de mudar para sinal quadrado e variar a frequência, mas provavelmente nao iamoss ver um sinal quadrado pois o espectro (infinito) deste teria que ser filtrado pelo anti-aliasing filter.

Vale a pena ir ao lab tirar esta foto? Resultados do loop??

3.2 BPSK

Demonstração dos Resultados usando como etapas as varias perguntas do enunciado, complementar com as fotos e possiveis tabelas ou partes de codigo

3.2.1 P1. Oscilador controlado numericamente

3.2.2 P2. Transmissor BPSK

O objectivo deste projeto é criar um transmissor BPSK com recurso a três elementos principais, uma fonte de bits, um codificador diferencial e mapeador, e um modulador. Neste projeto foi utilizada uma frequência de amostragem $f_s = 16$ kHz e uma frequência portadora $f_0 = 4$ kHz.

Para ter uma fonte de bits no transmissor usa-se um "bit-rate clock"cuja função vai ser criar uma sequência de bits b_n que a cada 16 ciclos gera um novo bit alternado, usando um contador com $f_s/16$ para determinar quando gerar um novo bit. Para alternar o bit basta negar o bit anteriormente obtido, tendo sido concretizado através de uma simples XOR:

$$b_n = b_{n-1} \oplus 1 \quad (1)$$

Após obter a fonte de bits passou-se ao segundo elemento do transmissor: o codificador diferencial e mapeador. Começando pelo codificador diferencial, este utiliza b_n para aplicar a seguinte operação lógica:

$$c_n = c_{n-1} \oplus b_n \quad (2)$$

Assim, com c_0 inicializado a zero codifica-se a sequência de bits b_n . Ao gerar b_n e c_n obtém-se dois sinais que variam entre "0"e "1"só que c_n tem o dobro do período (figura X).

Depois de obter c_n passa-se ao mapeamento do mesmo,

Falta agora gerar a onda portadora a modular. Esta foi implementada de forma mais simples face ao projeto anterior uma vez que a frequência é estática (4 kHz) e este valor consiste numa fração inteira da frequência de amostragem (16kHz). Em primeiro lugar criou-se uma tabela com quatro valores dum período da senoide, sendo esta:

Escolheram-se estes valores uma vez que o período de amostragem coincide com os instantes de máximo, mínimo e zero-crossing da portadora. Para gerar a portadora recorreu-se a um contador que,

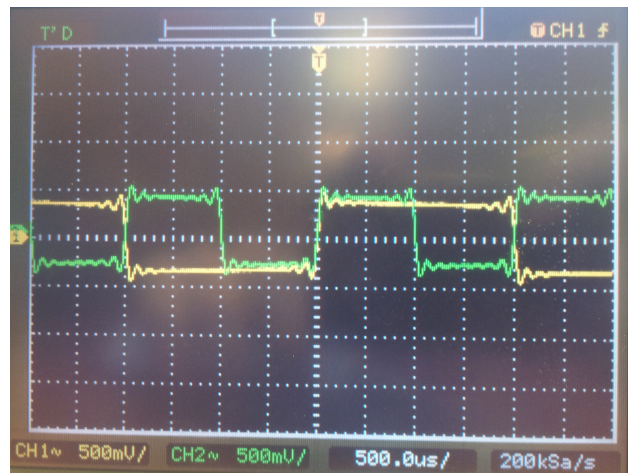


Figura 1: b_n (verde) e c_n (amarelo)

em cada interrupção (ocorrendo em cada instante de amostragem, como explicado no enunciado), aponta para cada entrada da tabela e põe a amostra numa variável que, após se incrementar o contador, irá ser multiplicada por d_n .

Tem-se dois sinais Q15, ou seja o bit mais significativo para o sinal e 15 bits para a parte fracionária. Ao multiplicar-se dois sinais Q15 sabe-se que o resultado será sempre Q15, contudo como o multiplicador retorna um valor em Q30, para este ser armazenado tem que se fazer shift left uma vez para eliminar o sinal repetido e shift right 16 vezes para transportar os bits mais significativos encostar os bits do resultado nos bits menos significativos para se truncar

Modulação=> tempo: é multiplicação da onda quadrada com a senoide. No espectro: as harmonicas ímpares (espectro quadrada) convoluídas com dirac a 4kHz(espectro portadora). As imagens que temos retratam a onda quadrada porque não temos 4 picos como deveria ser (o espectro transladado para +- 4khz). matlab na dropbox com o espectro correcto.

4 Conclusão

-Principais resultados e conclusões sobre eles, erros a corrigir (se houverem), o que melhorar

5 Anexos

-Codigo?

-possivelmente poe-se aqui algumas das imagens