

## УКАЗАНИЯ

за изпълнение на заданието от втората част на проекта по дисциплината  
„Мобилни и безжични комуникации“

В съответствие с поставените задачи в първата част на проекта по дисциплината „Мобилни безжични комуникации“ всеки студент разработва симулационен модел на комуникационна система със зададен тип на цифровата модулация, определени параметри на канала за връзка и зададен вид на цифровата информация за предаване. Целта на втората част на проекта по МБК е да се разшири създадения симулационен модел и студентите да придобият познания и умения, изпълнявайки следните задачи във втората част на проекта:

### **Задача № 1.**

Да се добавят в симулационния модел (от първата част на проекта) при разработката на втората част на заданието симулационни блокове за кодиране и декодиране на канала с цел за откриване и корекция на грешки. Да се изследва влиянието на добавените блокове за намаляване на грешките в приемната част на симулирания вид комуникационна система при запазване на същия по вид цифрова информация - случайна двоична последователност от данни, като резултатите се сравнят с тези, получени в първата част на проекта.

### **Задача № 2.**

Да се промени вида на предаваната цифрова информация от използваната в първата част на проекта случайна двоична последователност от данни в предаване на реален поток от избрани тестови аудио сигнали или тестови изображения.

### **Задача № 3.**

Предаването на реален поток от избрани тестови аудио сигнали или тестови изображения да се извърши без и с наличие на блокове за кодиране и декодиране на канала с цел откриване и корекция на грешки, като се сравнят резултатите за грешките в приемната част на симулирания вид комуникационна система, без и с наличие на блокове за кодиране и декодиране на канала с цел откриване и корекция на грешки.

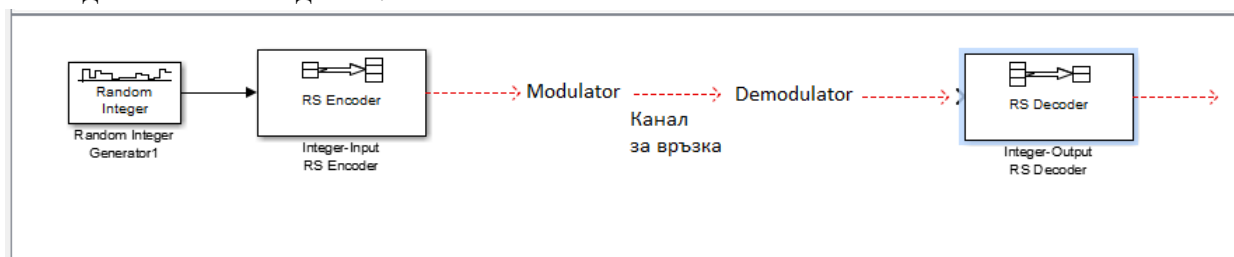
### **Задача № 4.**

Да се изследва предаването на реален поток от избрани тестови аудио сигнали или тестови изображения без и с наличие на блокове за кодиране и декодиране на канала с цел откриване и корекция на грешки, относно подобряване на качеството на приетите аудио сигнали или изображения. За целта в симулационния модел да се включат съответни симулационни блокове на възпроизвеждащи устройства - високоговорител или видео екран, чрез които слухово или съответно визуално да се оцени качеството на приетите тестове аудио сигнали или изображения.

Основната цел във втората част на проекта по МБК е да се симулира кодиране на канала с цел откриване и корекция на грешки, възникнали в комуникационния канал чрез добавяне в предаващата част блок за кодиране на канала след блока за генериране на случайна двоична последователност от данни и преди блока за модулация. Откриването и корекцията на грешките, възникнали при предаване на информацията по канала, се извършва чрез добавянето в приемната част на блок за декодиране на канала след блока за демодулация, например както това е показано на фиг.1.

Необходимо е да се подчертае, че в зависимост от зададения вид на блока за кодиране на канала с цел откриване на грешки следва да се променят неговите

параметри, а също така и параметрите на блока за генериране на случайна двоична последователност от данни.

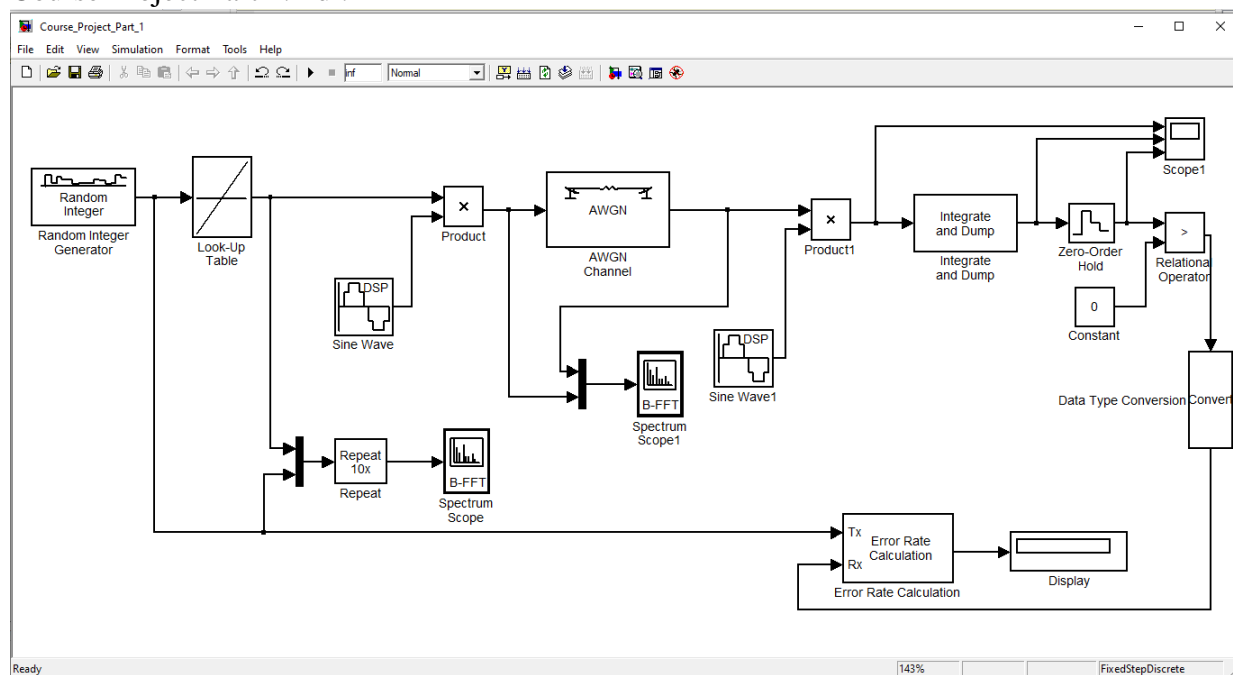


Фиг.1.

При изпълнението на втората част на проекта, съгласно на посочените по-горе изисквания, следва да се съблюдават следните по-важни указания:

### Указания при изпълнение на задача № 1.

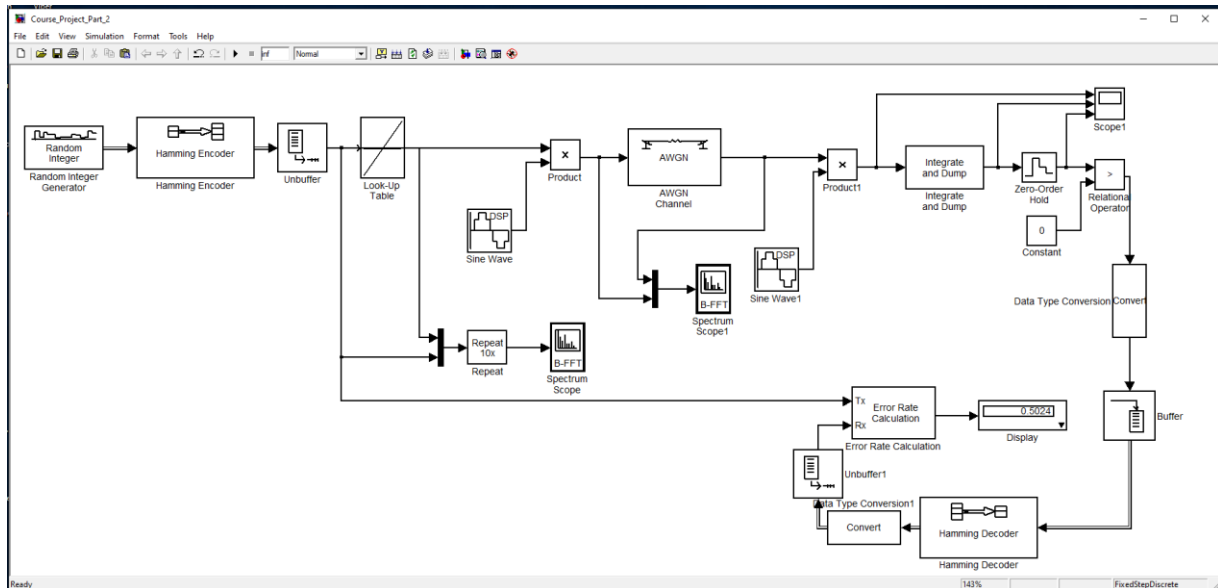
1. Създаденият симулационен модел в първата част на курсовия проект (фиг. 2), проверен че работи правилно, се записва с ново име. Например, ако името на курсовия проект от първата част е Course Project Part 1.mdl, се записва като друг файл с име Course Project Part 2.mdl.



Фиг.2. Примерен вид на правилно работещият модел от първата част на курсовия проект (зависи от конкретното задание за първата част на курсовия проект)

2. Отваря се в Simulink записания с ново име Course Project Part 2.mdl модел. Независимо от различията, в зависимост от конкретното задание, във видът на модела от първата част на курсовия проект, първият блок в Simulink модела във всички варианти е от вида Random Integer Generator. Затова, винаги първото действие при изпълнение на задача №1 от втората част на проекта, е да се прибави, в зависимост от конкретното задание за втората част на курсовия проект, след блока Random Integer Generator съответния блок от библиотеката на Simulink (Simulink Library Browser),

например: Hamming Encoder, Binary Linear Encoder, Integer-Input RS Encoder или др. Входът на прибавения блок кодер се свързва с първия блок Random Integer Generator, например, както е показано на Фиг. 3 за случай на задание с блок кодер Hamming Encoder.

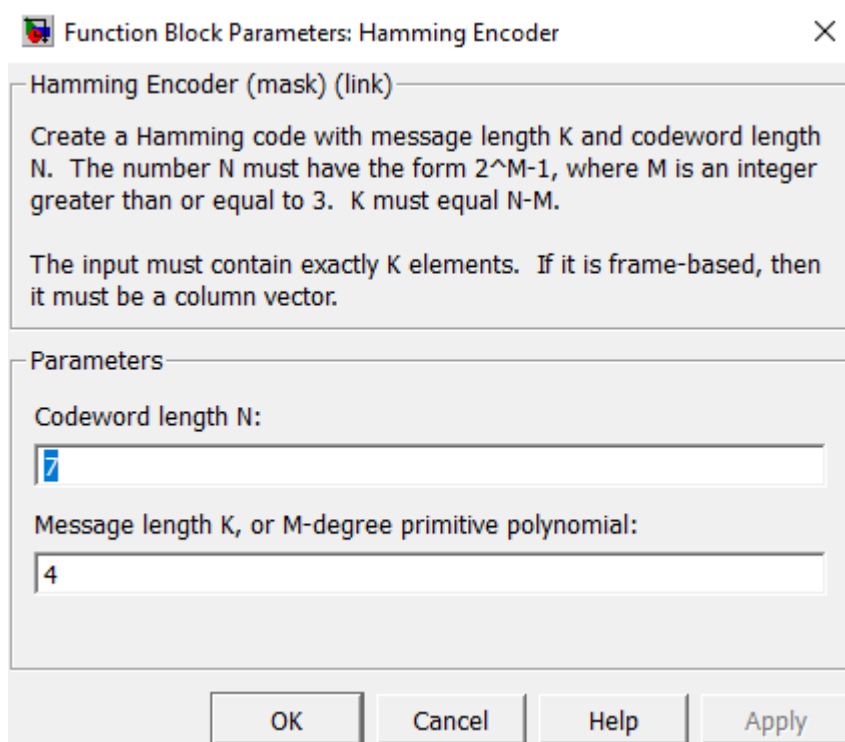


Фиг.3.

3. Съгласуване на параметрите на първия блок Random Integer Generator с параметрите на прибавения блок на кодер: Hamming Encoder, Binary Linear Encoder, Integer-Input RS Encoder или др., в зависимост от конкретното задание. За целта се изпълнява следното:

- с двукратно кликуване с мишката върху блока кодер: Hamming Encoder, Binary Linear Encoder, Integer-Input RS Encoder или др., в зависимост от конкретното задание, се отваря прозорец с параметрите на блока кодер (например, както е показано на Фиг. 4 за случай на задание с блок кодер Hamming Encoder);

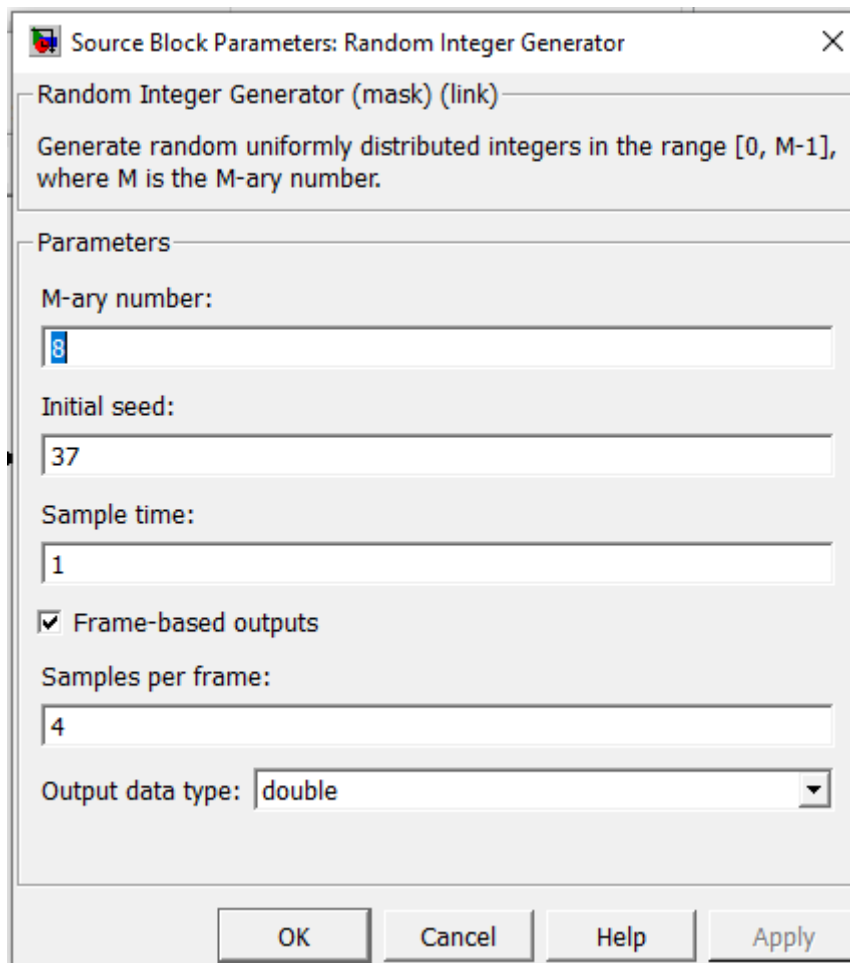
- параметрите Codeword length N (в конкретния пример на фиг.4 Codeword length N=7) и Message length K (в конкретния пример на фиг.4 Message length K=4) са съответно зададените стойности на дължината на кодовата дума Codeword length N на изхода на блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder) и дължината на съобщението Message length K на входа на блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder), т.е. на изхода на блока Random Integer Generator;



Фиг. 4. Отваряне с двукратно кликуване с мишката върху блока кодер на прозорец с параметрите на блока кодер, например за случай на задание с блок кодер Hamming Encoder

- дължината на съобщението Message length K (в конкретния пример на фиг.4 Message length K=4) на входа на блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder), следва да се съгласува с изходните параметри на предшестващия блок блока Random Integer Generator, т.е. да се зададе като изходен параметър на блока Random Integer Generator дължина на съобщението Message length K=4 (за конкретния пример на фиг. 4), като за целта с двукратно кликуване с мишката върху блока Random Integer Generator се отваря прозорец с параметрите на този блок, както е показано за пример на фиг.5, а в този прозорец следва да се променят, спрямо първата част на курсовия проект параметрите, както следва: активира се параметъра Frame-based outputs (изходният цифров поток е структуриран не по един бит, а в кадър „Frame“ по четири бита Message length K=4), а след това се променя стойността Samples per frame от 1 на 4, за конкретния пример на фиг.5.

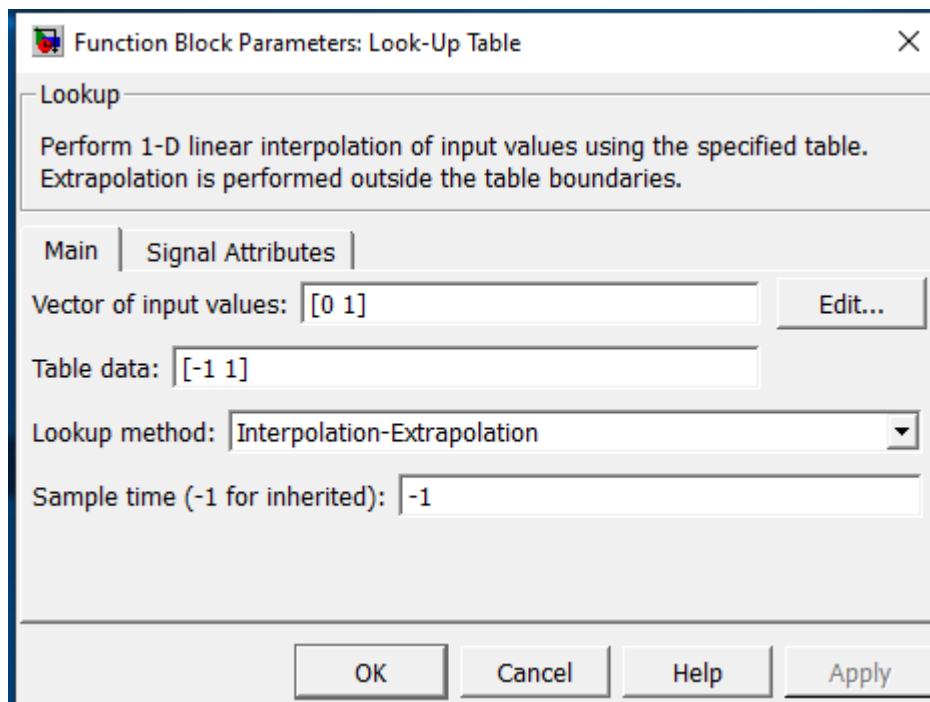
4. Необходимо е да се извърши съгласуване на изходните параметри Codeword length N на прибавения блок на кодер: Hamming Encoder, Binary Linear Encoder, Integer-Input RS Encoder или др., в зависимост от конкретното задание, с входните параметри на следващия, съществуващ и в модела Course Project Part 1.mdl от първата част на курсовия проект (фиг. 2), блок Look-up Table. Например, за конкретния примерен случай от фиг.4 се вижда, че изходните параметри на избрания кодер Hamming Encoder са Codeword length N=7, а входната информация или входните параметри на следващия блок Look-up Table следва да бъдат 1 bit Vector of input values [0 1]. За конкретния примерен случай от фиг.4 тези входни параметри на блока Look-up Table могат да се установят чрез двукратно кликуване с мишката върху блока Look-up Table (виж на фиг. 6).



Фиг. 5. Отваряне с двукратно кликване с мишката върху блока Random Integer Generator на прозорец за съгласуване (промяна) на неговите изходни параметри с входните параметри на блока кодир, например за случай на задание с блок кодир Hamming Encoder

5. За да се осъществи описаното по-горе съгласуване, следва в модела Course Project Part 2.mdl след кодера (на фиг. 3 след Hamming Encoder) да се прибави блок от типа Unbuffer, който да се свърже с входа на блока Look-up Table. Чрез блокът Unbuffer дължината на кодовата дума Codeword length от N bits в изхода на кодера се преобразува в 1 bit. За примера дължината на кодовата дума в изхода на избрания кодир Hamming Encoder е Codeword length N=7 bits и се преобразува от блока Unbuffer в един bit.

6. След прибавянето на блока Unbuffer, между кодера (за примера на фиг.3 Hamming Encoder) и блока Look-up Table, е необходимо да се промени връзката към блоковете Repeat и Error Rate Calculation (към входа Tx). Тази връзка, в модела от първата част на курсовия проект Course Project Part 1.mdl (за примера на фиг.2), е от изхода на блока Random Integer Generator, а в модела от втората част на курсовия проект Course Project Part 2.mdl (за примера на фиг.3) следва да се промени и да бъде от изхода на блока Look-up Table.



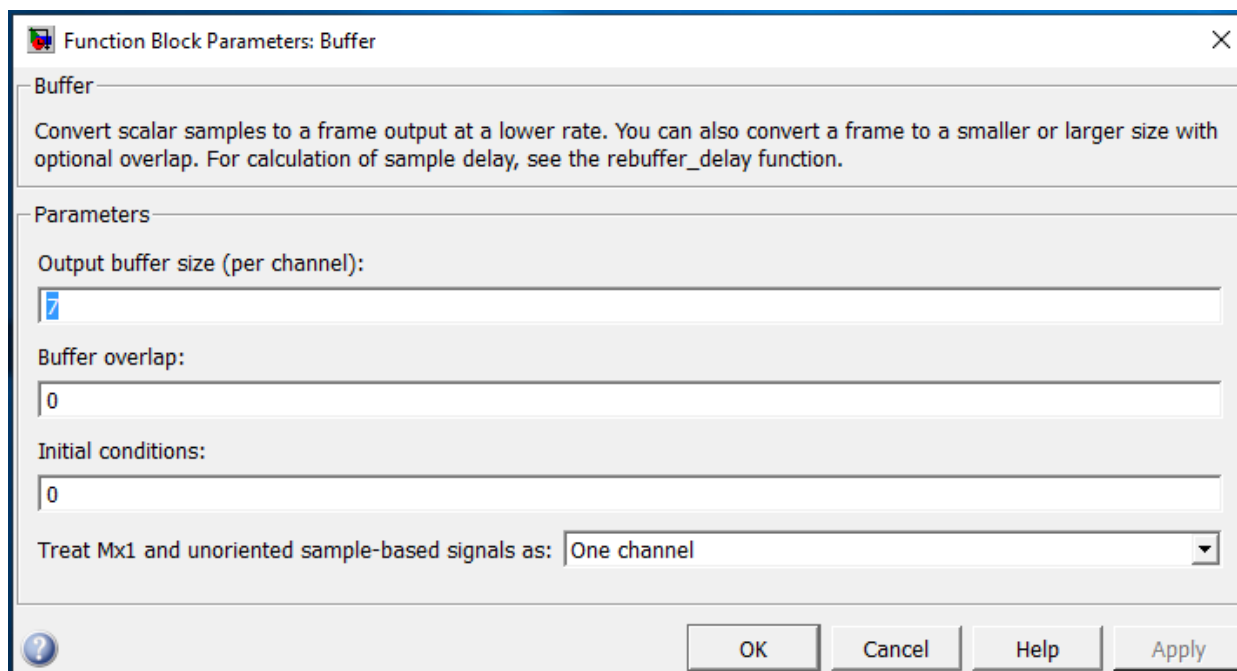
Фиг.6. Входната информация или входните параметри на блока Look-up Table могат да се установят чрез двукратно кликуване с мишката върху блока Look-up Table и за конкретния примерен случай от фиг.4 са 1 bit и се определят от стойностите в полето Vector of input values, в случая [0 1]

7. Описаните по-горе действия, при изпълнение на задача №1 от втората част на проекта, се отнасят до предаващата (кодираща) част в модела от втората част на курсовия проект Course Project Part 2.mdl (за примера на фиг.3). Така формираното кодирано съобщение или кодираната информация се предава по канала за връзка (AWGN Channel за примера на фиг.3). След приемането на кодирано съобщение или кодираната информация следва да се извършат подобни действия и в приемната (декодираща) част в модела от втората част на курсовия проект Course Project Part 2.mdl (за примера на фиг.3). Първото действие в този случай е да се прибави след блока Data Type Conversion блок на декодер, който в зависимост от конкретното задание за втората част на курсовия проект може да бъде, например: Hamming Decoder, Binary Linear Decoder, Integer-Input RS Decoder или др. Съответния блок на декодер следва да се потърси от библиотеката на Simulink (Simulink Library Browser), например: Hamming Decoder, Binary Linear Decoder, Integer-Input RS Decoder или др. Входът на прибавения блок на декодер се свързва с изхода на блока Data Type Conversion, например, както е показано на Фиг.3 за случай на задание с блок декодер Hamming Decoder, а изходът на декодера (за примера от фиг. 3 Hamming Decoder) да се свърже към входа Rx на блока Error Rate Calculation.

8. Прибавянето само на блок декодер (както е показано на Фиг.3 за случай на задание с блок декодер Hamming Decoder) след блока Data Type Conversion не е достатъчно и би довело до грешки при стартиране на модела, тъй като форматът на данните на изхода на блока Data Type Conversion и форматът на данните на входа на блока декодер (на Фиг. 3 за случай на задание с блок декодер Hamming Decoder) не е еднакъв. За целта е необходимо да се извърши допълнително следното съгласуване на формата на данните между тези два блока в модела от втората част на курсовия проект Course Project Part 2.mdl (за примера на фиг.3):

- между изхода на блока Data Type Conversion и входа на блока декодер (на Фиг. 3 за случай на задание с блок декодер Hamming Decoder) следва да се прибави блок с наименование Buffer, който трябва да се потърси в библиотеката на Simulink (Simulink Library Browser);

- параметрите на прибавения блок Buffer следва да се зададат, както е показано на фиг.7, за разглеждания случай на модел от втората част на курсовия проект Course Project Part 2.mdl показан на фиг.3. Дължината на кодовата дума Codeword length от N bits на входа на блока декодер Hamming Decoder се задава Codeword length N=7. Така съответства на дължината на кодовата дума в изхода на избрания кодер Hamming Encoder - Codeword length N=7;



Фиг.7. Задаване на параметрите на прибавения блок Buffer, например за разглеждания случай на модел от втората част на курсовия проект Course Project Part 2.mdl показан на фиг.3, съответната дължина на кодовата дума Codeword length от N bits на входа на блока декодер Hamming Decoder трябва да се зададе Codeword length N=7 и да отговаря на дължината на кодовата дума в изхода на избрания кодер Hamming Encoder - Codeword length N=7

9. След извършване на описаното по-горе съгласуване на формата на данните между изхода на блока Data Type Conversion и входа на блока декодер (на Фиг. 3 за случай на задание с блок декодер Hamming Decoder) чрез прибавяне на блок Buffer между тях, в изхода на блока декодер формата на данните е с дължина на декодираното съобщение Message length K, т.е съответства на дължината Message length K (в конкретния пример на фиг.4 Message length K=4) на съобщението или информация преди кодиране в блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder). Освен с дължина Message length K, декодираното съобщение в изхода на блока декодер (на Фиг. 3 за случай на задание с блок декодер Hamming Decoder) се характеризира и с точност на формата на данните, който в Matlab за блок декодер (в случая за примера Hamming Decoder) е формат с двойна точност (double precision). Затова е необходимо към изхода на блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder) да се прибави блок Data Type Conversion,

както е показано на фиг.3. Чрез него формата на данните с двойна точност (double precision) в изхода на блока кодер (на Фиг. 4 за случай на задание с блок кодер Hamming Encoder) се преобразува във формат на цели числа без знак за плюс и минус (в Matlab се означава като uint8 или 8-bit unsigned integer). След преобразуването на формата на данните като uint8, чрез прибавения блок Data Type Conversion, дължината Message length K остава същата (в конкретния пример на фиг.4 Message length K=4), както е на изхода на блока декодер (на Фиг. 3 за случай на задание с блок декодер Hamming Decoder). Следва да се извърши преобразуване на тази дължина чрез прибавяне след блока Data Type Conversion на блок Unbuffer и изхода на този блок да се свърже към входа Rx на блока Error Rate Calculation, чрез което се осъществява съгласуване с формата на данните (1 bit) на входа Rx на блока Error Rate Calculation с формата на данните (1 bit) на входа Tx на същия блок.

**Забележка:** Изпълнението на описаните по-горе действие (указания за изпълнение на задача № 1 от втората част на курсовия проект), зависят от конкретното задание за курсовия проект (Hamming Encoder/Decoder, Binary Linear Encoder/Decoder, Integer-Input RS Encoder/Decoder или др.). Възможно е, при съответните конкретни случаи, някои от описаните действия да не са необходими или да се наложи включване на подобни на описаните по-горе допълнителни блокове, например допълнителни блокове: блок Unbuffer, блок Buffer, блок (Data Type Conversion) или други подобни блокове, главно за съгласуване на формата на данните между съществуващите блокове в модела от първата част на курсовия проект Course Project Part 1.mdl и добавяните в модела от втората част на курсовия проект Course Project Part 2.mdl (Hamming Encoder / Decoder, Binary Linear Encoder / Decoder, Integer-Input RS Encoder / Decoder или др.).