

Задание 1. Построить структурную схему и написать код, описывающий формулу ниже, выполнить проверку.

$$ind = mod(f1 * i + f2 * i^2, K),$$

где ind - выходной вектор от 0 до K,

f1, f2, K входные числа, заданные в таблице для проверки, константы в пределах выбранного массива.

i - вектор от 0 до K.

	f1	f2	K
check_48	7	12	48
check_1008	55	84	1008
check_6144	263	480	6144

Формула представляет собой перемежитель по стандарту LTE (LTE Turbo code interleaver). Выводит индексы данных из блока для перемешивания.

На вход подаётся массив чисел

Для проверочного массива check_48: i = меняется от нуля до 47 с шагом 1

Для проверочного массива check_1008: i = от нуля до 1007 с шагом 1

Для проверочного массива check_6144: i = от нуля до 6143 с шагом 1

Таким образом, необходимо реализовать математику, выделения остатка из деления на K.

Требуемый результат выполнения тестового задания: Структурная схема алгоритма, код на xHDL, результаты верификации по высланным данным.

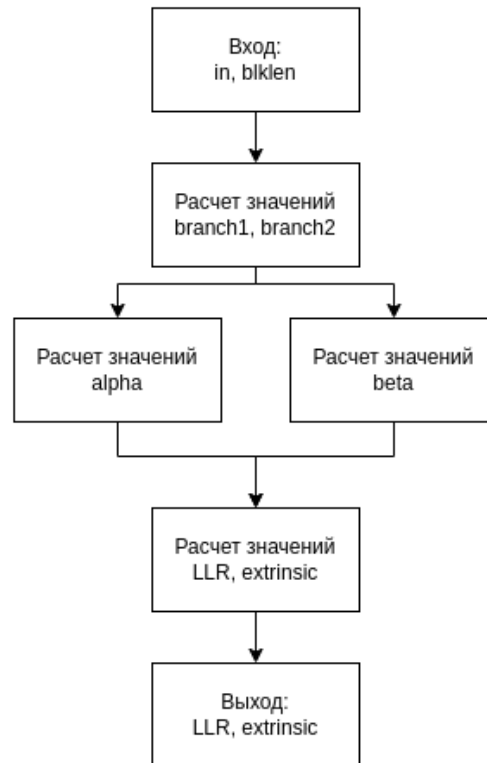
Предложения по оптимизации приветствуются.

Данные для проверки приведены в отдельных файлах.

Задание 2.

Данный алгоритм представляет собой SISO декодер, являющийся подмодулем Турбо декодера по стандарту LTE (по алгоритму Max-Log-MAP).

На вход поступает массив данных и длина массива. Общая структурная схема обработки имеет следующий вид:



Входной массив `in` состоит из целых чисел типа `int16`. Каждый второй элемент с начального переписывается в массив `sys`, каждый второй элемент с начального+1 элемента переписывается в массив `parity`. Далее проходит расчёт данных `branch1` и `branch2`. Затем рассчитываются значения в массив `alpha` и `beta`, используя результаты расчётов `branch1/branch2`. В конечном итоге, по имеющимся данным, рассчитывается `LLR` и `extrinsic`.

Требуемый результат выполнения тестового задания: Необходимо оценить срок разработки и/или написать алгоритм на xHDL, оценить потребляемые ресурсы, верифицировать по высланным данным. Алгоритм представлен на схеме ниже. Данные для проверки в отдельном файле.

Read:
in

initialize:
 $\alpha(0) = [0; -128; -128; -128; -128; -128; -128; -128]$
 $\beta(0) = [0; -128; -128; -128; -128; -128; -128; -128]$
 $\text{sys} = \text{in}(0:2:\text{end})$
 $\text{parity} = \text{in}(1:2:\text{end})$

Calculate:
 $\text{branch1} = -(\text{sys} + \text{parity})/2$
 $\text{branch2} = -(\text{sys} - \text{parity})/2$

$k = 1$

$k < \text{length}(\text{in})?$
False
True

Calculate:
 $\alpha(0,k) = \max(\alpha(0,k-1) + \text{branch1}(k-1), \alpha(1,k-1) - \text{branch1}(k-1))$
 $\alpha(1,k) = \max(\alpha(2,k-1) - \text{branch2}(k-1), \alpha(1,k-1) + \text{branch2}(k-1))$
 $\alpha(2,k) = \max(\alpha(4,k-1) + \text{branch2}(k-1), \alpha(5,k-1) - \text{branch2}(k-1))$
 $\alpha(3,k) = \max(\alpha(6,k-1) - \text{branch1}(k-1), \alpha(7,k-1) + \text{branch1}(k-1))$
 $\alpha(4,k) = \max(\alpha(0,k-1) - \text{branch1}(k-1), \alpha(1,k-1) + \text{branch1}(k-1))$
 $\alpha(5,k) = \max(\alpha(2,k-1) + \text{branch2}(k-1), \alpha(3,k-1) - \text{branch2}(k-1))$
 $\alpha(6,k) = \max(\alpha(4,k-1) - \text{branch2}(k-1), \alpha(5,k-1) + \text{branch2}(k-1))$
 $\alpha(7,k) = \max(\alpha(6,k-1) + \text{branch1}(k-1), \alpha(7,k-1) - \text{branch1}(k-1))$

Normalize:
 $\alpha(:,k) = \alpha(:,k) - \alpha(0,k)$

$k = k + 1$

$k = \text{length}(\text{in}) - 1$

$k < 0?$
False
True

Calculate:
 $\alpha(0,k) = \max(\beta(0,k+1) + \text{branch1}(k), \beta(4,k+1) - \text{branch1}(k))$
 $\alpha(1,k) = \max(\beta(4,k+1) - \text{branch1}(k), \beta(0,k+1) - \text{branch1}(k))$
 $\alpha(2,k) = \max(\beta(5,k+1) + \text{branch2}(k), \beta(1,k+1) - \text{branch2}(k))$
 $\alpha(3,k) = \max(\beta(1,k+1) + \text{branch2}(k), \beta(5,k+1) - \text{branch2}(k))$
 $\alpha(4,k) = \max(\beta(2,k+1) + \text{branch2}(k), \beta(7,k+1) - \text{branch2}(k))$
 $\alpha(5,k) = \max(\beta(6,k+1) + \text{branch2}(k), \beta(2,k+1) - \text{branch2}(k))$
 $\alpha(6,k) = \max(\beta(7,k+1) + \text{branch1}(k), \beta(3,k+1) - \text{branch1}(k))$
 $\alpha(7,k) = \max(\beta(3,k+1) + \text{branch1}(k), \beta(7,k+1) - \text{branch1}(k))$

Normalize:
 $\beta(:,k) = \beta(:,k) - \beta(0,k)$

$\text{LLR}(k) = \max(\alpha(0,k) - \text{branch1}(k) + \beta(4,k+1), \alpha(1,k) - \text{branch1}(k) + \beta(0,k+1), \alpha(2,k) - \text{branch2}(k) + \beta(1,k+1), \alpha(3,k) - \text{branch2}(k) + \beta(5,k+1), \alpha(4,k) - \text{branch2}(k) + \beta(6,k+1), \alpha(5,k) - \text{branch2}(k) + \beta(2,k+1), \alpha(6,k) - \text{branch1}(k) + \beta(3,k+1), \alpha(7,k) - \text{branch1}(k) + \beta(7,k+1))$
 $\max(\alpha(0,k) + \text{branch1}(k) + \beta(0,k+1), \alpha(1,k) + \text{branch1}(k) + \beta(4,k+1), \alpha(2,k) + \text{branch2}(k) + \beta(5,k+1), \alpha(3,k) + \text{branch2}(k) + \beta(1,k+1), \alpha(4,k) + \text{branch2}(k) + \beta(2,k+1), \alpha(5,k) + \text{branch2}(k) + \beta(6,k+1), \alpha(6,k) + \text{branch1}(k) + \beta(8,k+1), \alpha(7,k) + \text{branch1}(k) + \beta(3,k+1))$

$k = k - 1$

Calculate:
 $\text{extrinsic} = 0.75 * (\text{LLR} - \text{sys})$

Save (output):
LLR, extrinsic