Применение технологии CUDA для решения задач демодуляции сигналов

Докладчик:
Зубков Михаил Павлович

Закон Амдала

$$S = \frac{1}{(1-P) + \frac{P}{N}}$$

где P - это часть времени выполнения программы, которая может быть распараллелена на N процессоров

При $N \to \infty$

$$S \to \frac{1}{1-P} \quad \Rightarrow \quad \text{при } P \to 1 \quad S \to \infty$$

Технология CUDA

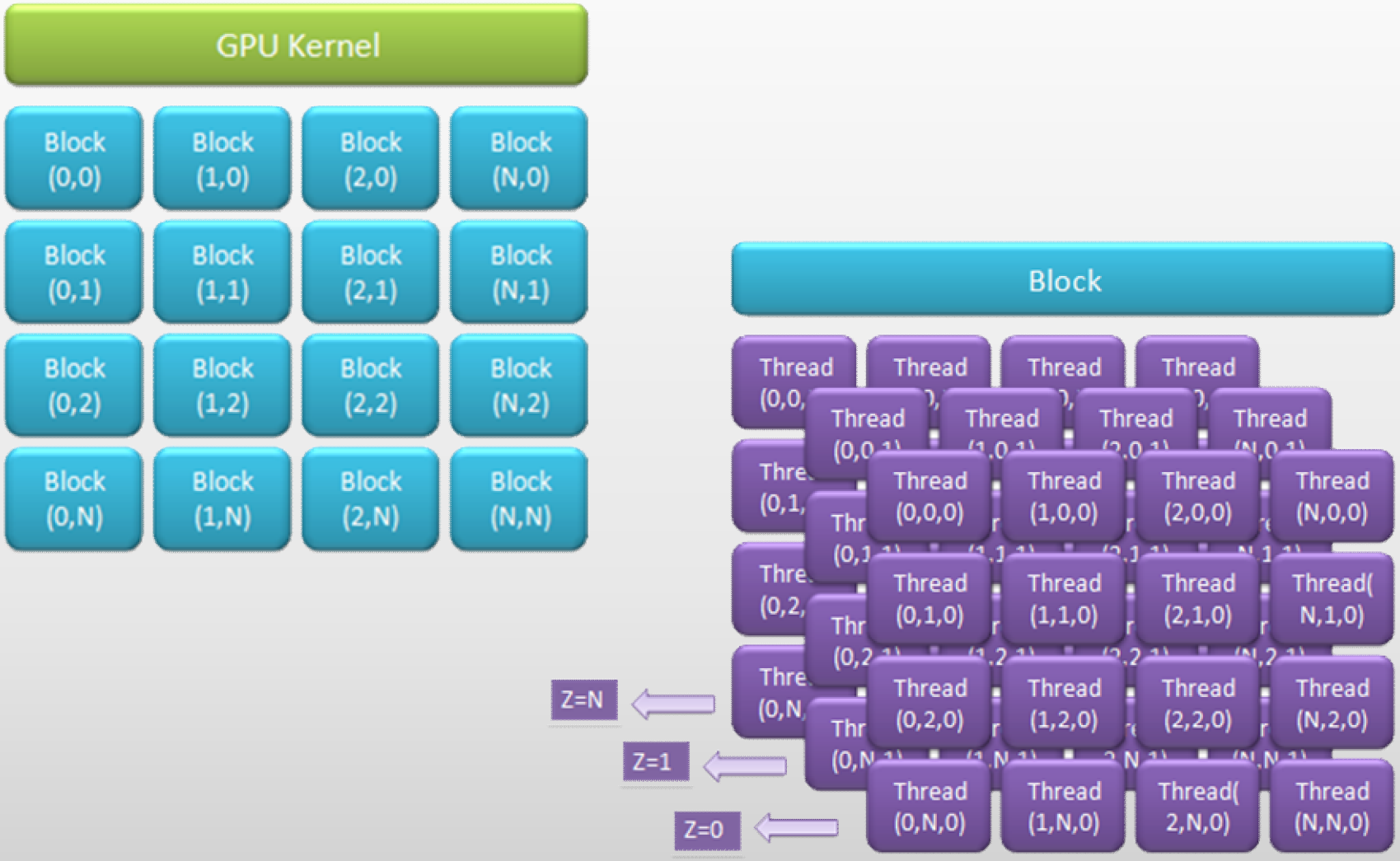
Преимущество:

- Свободная лицензия
- Кроссплатформенность
- Наличие подробной документации
- Наличие большого количества библиотек

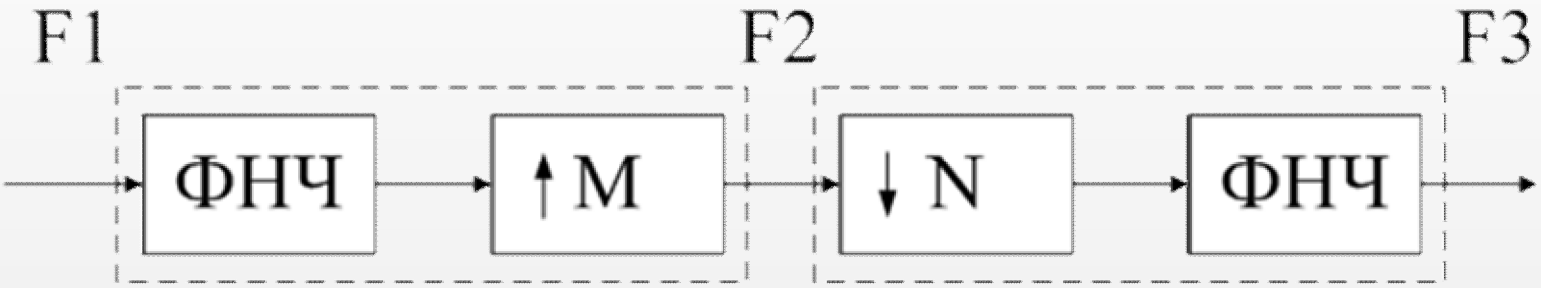
Применение:

- Ускорения вычислений в нейронных сетях
- Моделирование различных процессов и систем
- Обработка изображений и распознавания образов
- Кодирование и декодирование видео

Вычислительная модель GPU



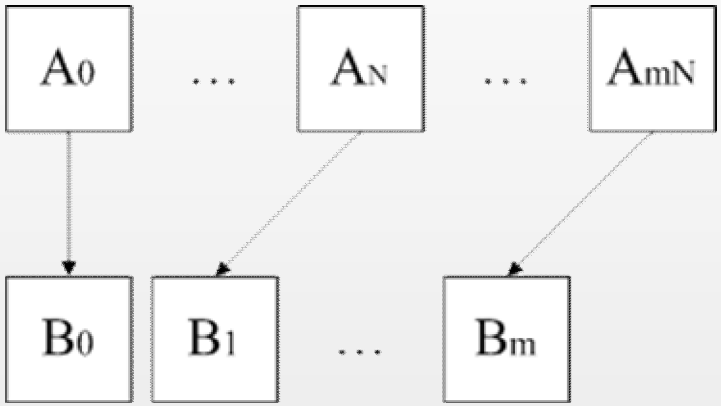
Структурная схема операции передискретизации



ФНЧ	Фильтр нижних частот
↓ N	Децимация
↑ M	Интерполяция

$$F_3 = F_1 \frac{M}{N}$$

Децимация



Последовательный код на CPU

```
void decimate(int *a, int * b, int N, int size)
{
    for(int i=0;i<size/N;i++) b[i]=a[N*i];
}
```

Ядро CUDA

```
__global__ void cudaDecimate(int *a,  int * b,  int N)
{
      int i=threadIdx.x+blockIdx.x*blockDim.x;
      b[i]=a[N*i];
}
```

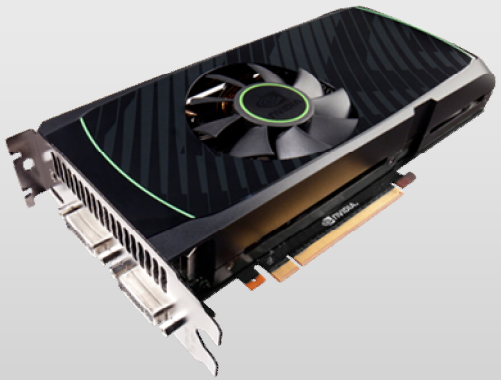
dim3 threadIdx - текущий индекс нити в блоке
dim3 blockIdx - текущий индекс блока в сетке
dim3 blockDim - размерность блока

Децимация – портирование кода на CUDA

```
void decimate(int *a, int *b, int N, int size)
{
	float *aDev;
	float *bDev;
	//Выделение памяти
	cudaMalloc((void **)&aDev, size*sizeof(float));
	cudaMalloc((void **)&bDev, size/N*sizeof(float));
	//Настройка конфигурации запуска
	dim3 threads = dim3(512,1);
	dim3 blocks = dim3((size/N)/threads.x,1);
	//Копирование входных данные из памяти CPU в память GPU
	cudaMemcpy(aDev, a,  size*sizeof(float), cudaMemcpyHostToDevice);
	//Вызов ядра с заданной конфигурации для обработки данных
	cudaDecimate<<<blocks,threads>>>(aDev, bDev, N);
	//Копирование выходных данные в памяти CPU
	cudaMemcpy(b, bDev, size/N*sizeof(float), cudaMemcpyDeviceToHost);
	//Освобождение выделенной память GPU
	cudaFree(aDev);
	cudaFree(bDev);
}
```

Характеристики процессоров

	Intel Core i7-2600	GTX 560 Ti
Число ядер	4	384
Частота работы ядра	3.4 ГГц	822 МГц
Пиковая производительность	57,8 ГФлопс	842,24 ГФлопс
Средняя стоимость	9000 руб.	6000 руб.



Исходные данные эксперимента

- Описание операции:

  Операция передискретизации с частоты 64 кГц на частоту 96 кГц (интерполяция на 3, фильтрация ФНЧ 78 порядка и последующая децимация на 2) ;

- Формат обрабатываемых данных:

  с плавающей запятой одинарной точности (float)

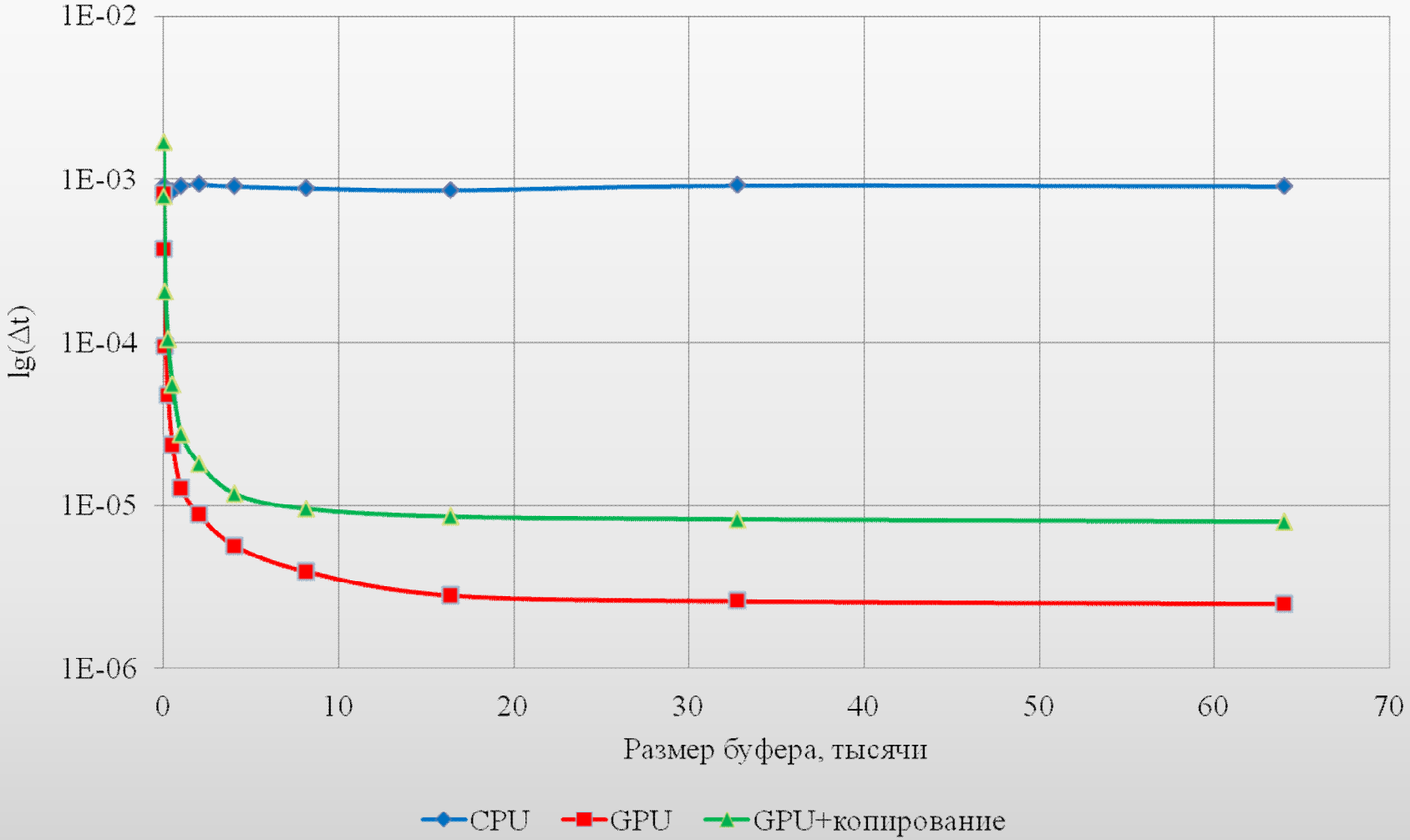
- Проводимые измерения:

  Время выполнения операции на ЦП, время выполнения операции на ГП, Время выполнения операции на ЦП с копированием данных с оперативно памяти в память ГП и обратно.

Сравнительная таблица выполнения операции передискретизации

Размер массива	Δt_{CPU}, мсек	Δt_{GPU}, мсек	$\Delta t_{GPU+копирование}$, мсек	$\Delta t_{CPU} / \Delta t_{GPU}$	$\Delta t_{CPU} / \Delta t_{GPU+копирование}$
16	0,014	0,013	0,027	1,08	0,52
32	0,029	0,012	0,025	2,42	1,16
128	0,11	0,012	0,026	9,17	4,23
256	0,22	0,012	0,027	18,33	8,15
512	0,43	0,012	0,028	35,83	15,36
1024	0,93	0,013	0,028	71,54	33,21
2048	1,9	0,018	0,037	105,56	51,35
4096	3,7	0,023	0,048	160,87	77,08
8192	7,2	0,032	0,078	200	92,31
16384	14	0,046	0,14	318,18	100
32768	30	0,085	0,27	352,94	111,11
64000	58	0,16	0,51	362,5	113,73

График зависимости времени обработки на один отсчет от размера входного буфера



Результаты измерения

- Время обработки одного отсчета на ЦП не зависит от общего размера обрабатываемого буфера.
- Время обработки одного отсчета на ГП уменьшается при возрастании общего размера обрабатываемого буфера до определенного значения, дальше время обработки практически не изменяется.
- При возрастании размера обрабатываемого буфера увеличивается выигрыш вычислений на ГП относительно вычисления на ЦП и при размере буфера в 64000 отсчетов составляет 362,50 без учета копирования данных и 113,73 с учетом копирования.

Исследуемые алгоритмы демодуляции сигналов

- Демодуляция сигнала частотно-временных матриц (ЧВМ)
- Демодуляция сигнала частотной телеграфии (ЧТ)
- Демодуляция сигнала относительной фазовой телеграфии (ОФТ)
- Демодуляция многоканального сигнала относительной фазовой телеграфии (МОФТ)

Формат обрабатываемого потока данных

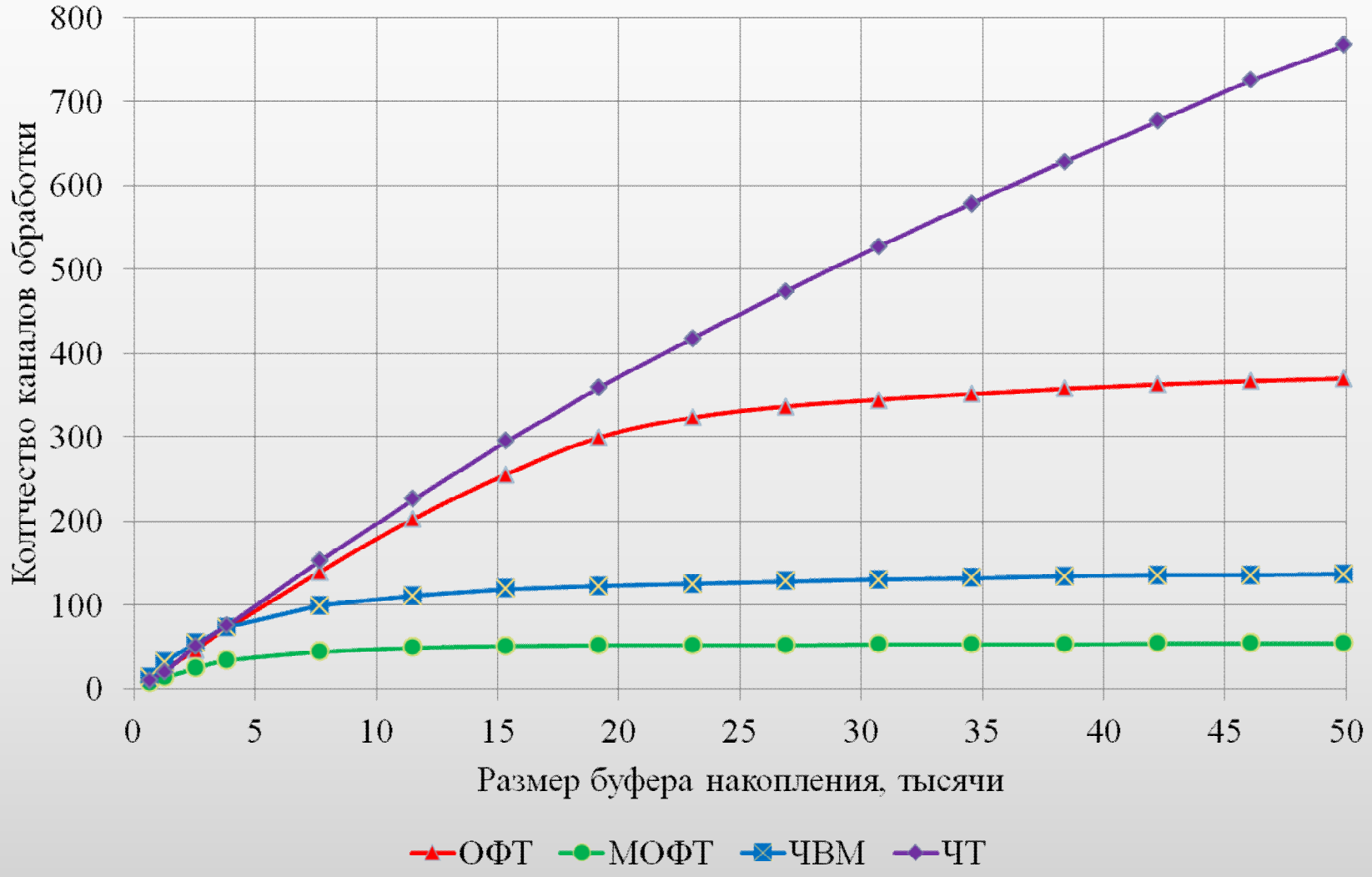
- Вид обрабатываемого сигнала:

  сигнал представляется в виде синфазной и квадратурной составляющих с частотой дискретизации 64 кГц

- Формат обрабатываемых данных:

  с плавающей запятой одинарной точности (float)

График зависимости количества каналов демодуляции сигнала от размера буфера накопления



Результаты измерения

- Максимальное количество обрабатываемых каналов демодуляции зависит от алгоритма демодуляции и размера входного буфера.
- Увеличение входного буфера влечет за собой увеличение времени реакции системы.
- Значение реакции системы, следовательно и количество каналов обработки, необходимо выбирается из условий решаемой задачи.
- Максимальное количество каналов обработки достигнуто при демодуляции ЧТ сигнала и равняется 767, при времени реакции системы 390 мсек.

Результаты измерения

- Оптимальное количество каналов обработки при демодуляции ОФТ сигнала равняется 323, при времени реакции системы 180 мсек.
- Оптимальное количество каналов обработки при демодуляции ЧВМ сигнала равняется 119, при времени реакции системы 120 мсек.
- Оптимальное количество каналов обработки при демодуляции МОФТ сигнала равняется 44, при времени реакции системы 60 мсек.

Спасибо за внимание!