

FIR单元参考设计

杜伟韬 duweitao@cuc.edu.cn

广播电视数字化工程中心

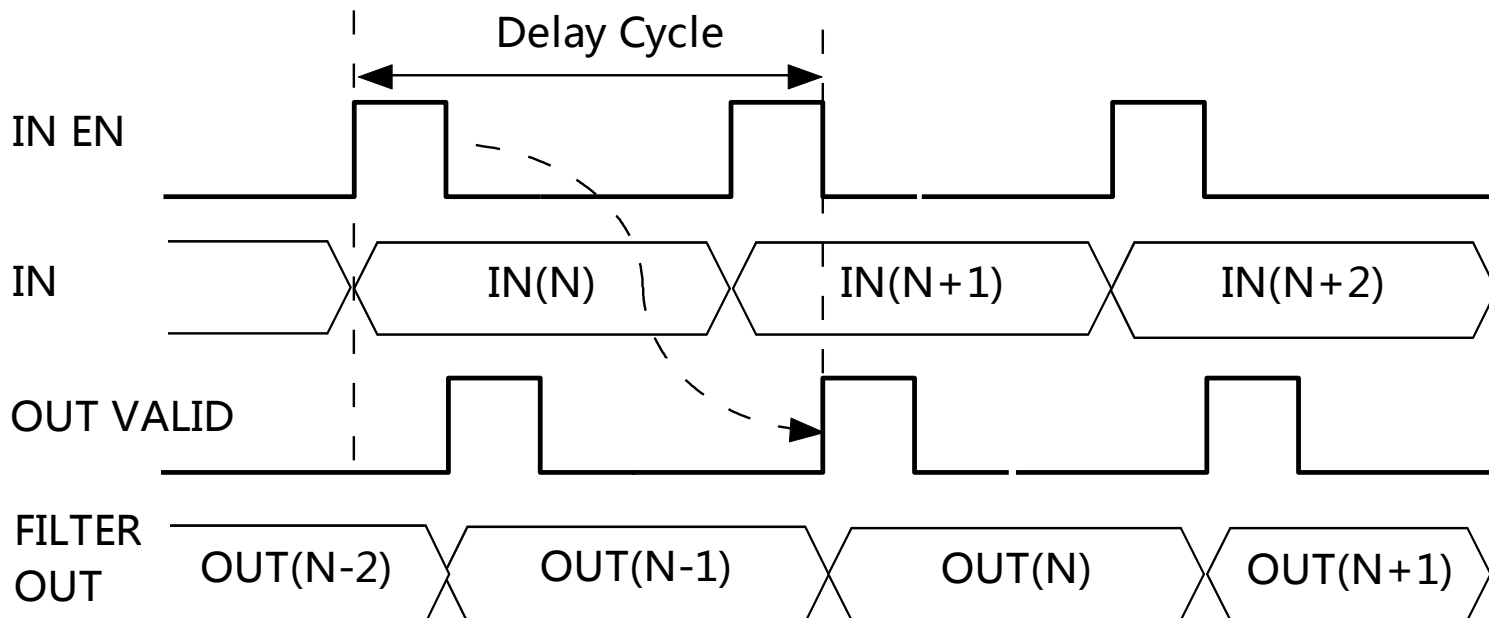
中国传媒大学

内容

- 设计需求
 - 时序、结构、重用和扩展
- 算法设计
 - 理论算法、标号/地址映射、定点策略
- 电路结构
 - 模块结构
 - 流水线设计
- 模块测试
 - Modelsim仿真测试
 - 电路板运行结果测试
- 设计、验证不同参数的滤波器
 - 激励模式，样点的时间间隔
 - 不同的频响
 - 系数个数，数据字长

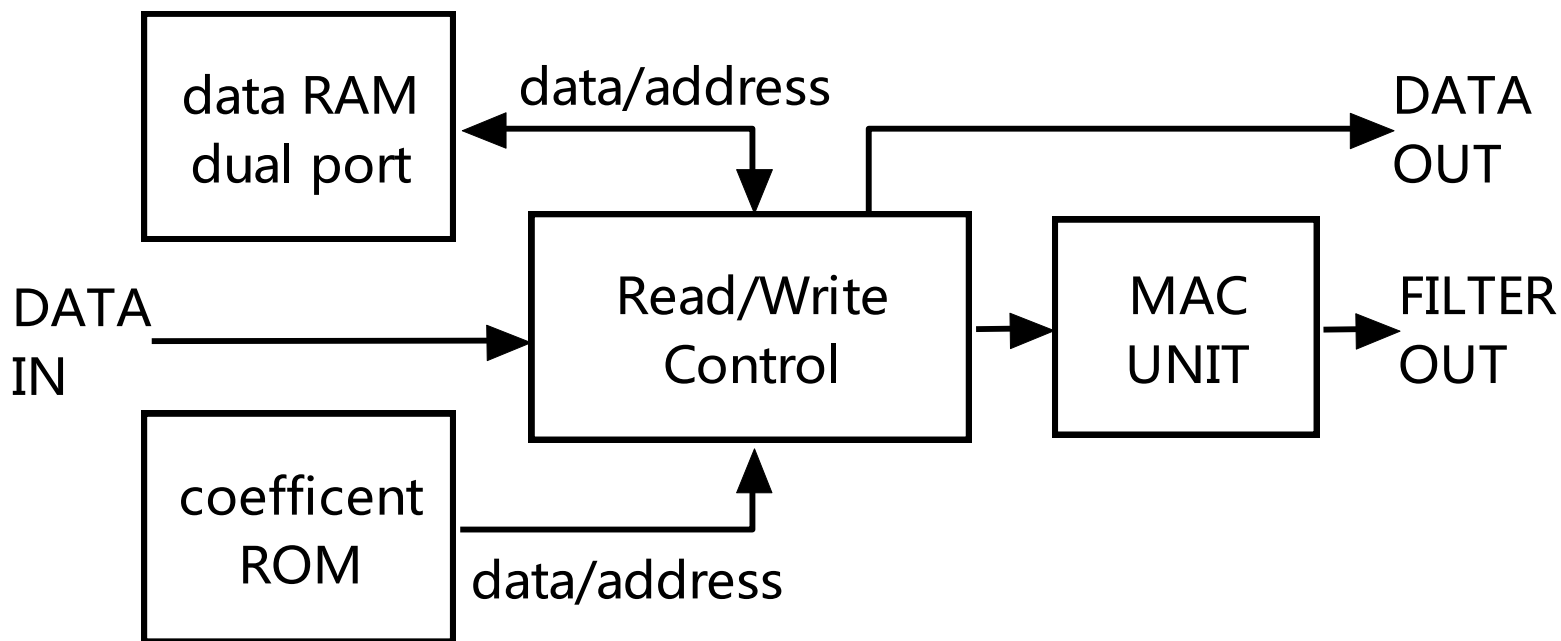
设计需求—时序

- 数据驱动—使用使能信号控制数据I/O
 - 设 N 为系数个数
 - 计算时间为 N 个时钟+流水延迟数目
 - 完成计算后输出结果并等待下一个数据输入
 - 输入数据间隔可以任意，但必须不小于 N 个时钟



设计需求—电路结构

- 使用单个乘法累加器 (MAC)
- 使用单口ROM作为系数存储器
- 使用双口RAM作为数据存储器

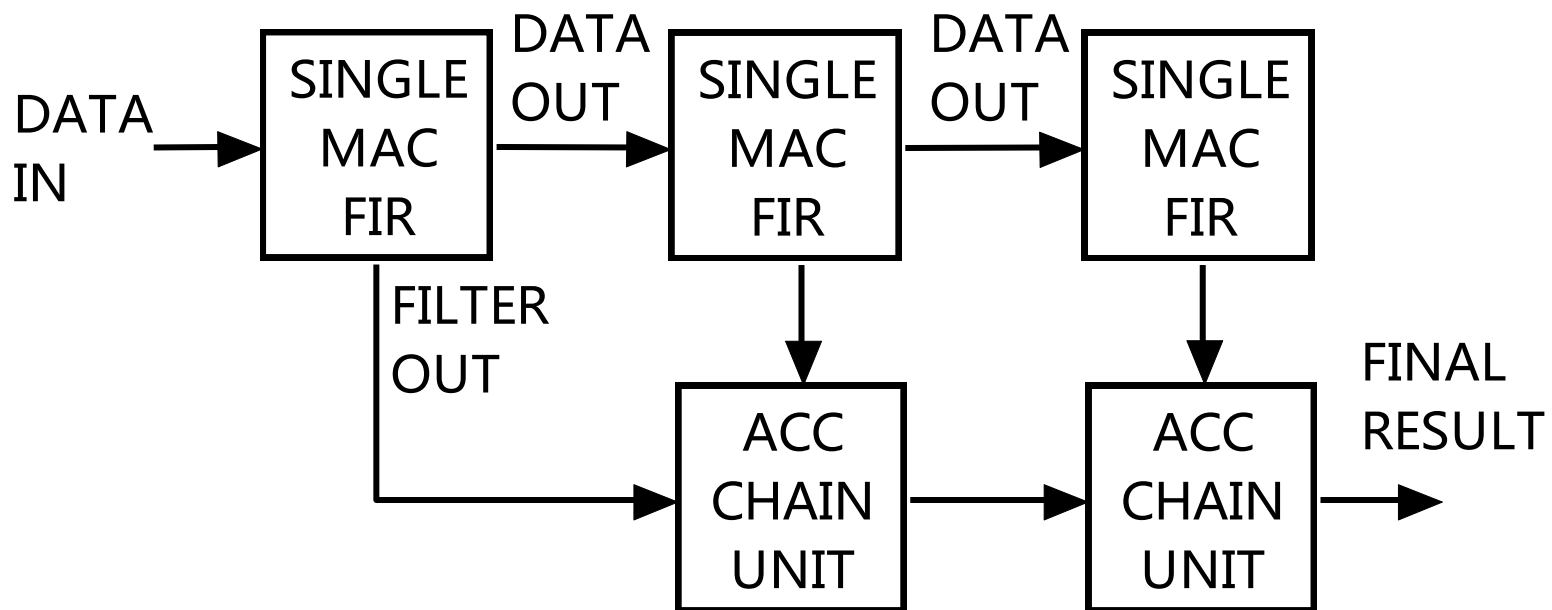


设计需求—重用和扩展

- 抽头系数改动
 - 使用Matlab生成系数ROM
 - 系数个数的参数化配置
- 数据字长
 - 输入字长参数化，输出字长按最大值推算
- 作为脉动阵列拼接单元
 - 带有移出数据的接口
- 作为自适应滤波器的滤波执行单元
 - 替换系数ROM为双口RAM

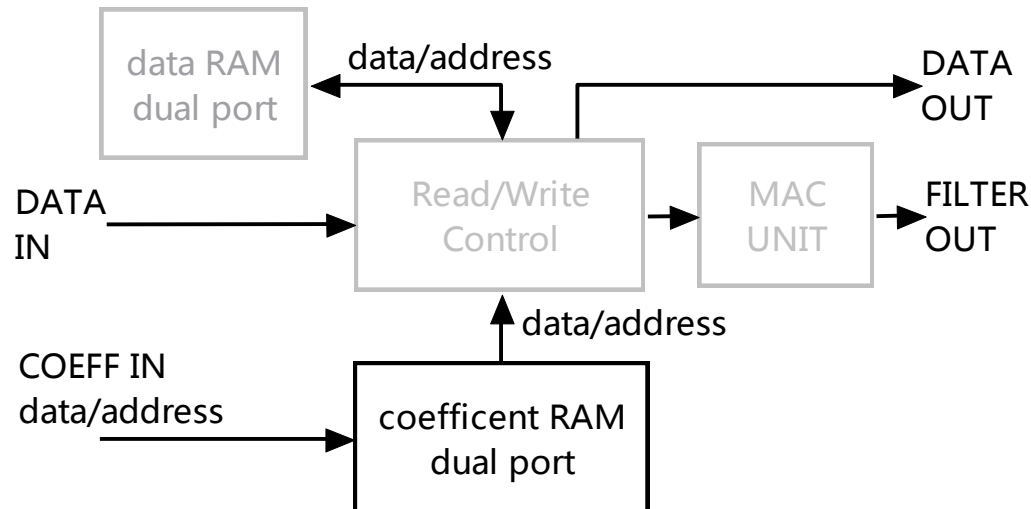
扩展—并发计算的脉动滤波阵列

- 级联单MAC的滤波器
- 当前脉动单元的输入数据来自前级的移出数据
- 构造累加链—级联累加器单元



扩展—自适应滤波器的滤波执行单元

- 替换系数ROM为双口RAM
- 外部电路提供写入数据、地址、使能信号
- 写入的系数和滤波器的输入数据之间，注意时序匹配关系
 - 注意避免破坏流水线
 - 例如第N个输入数据D(N)使用系数矢量C1进行滤波
 - 第N+1个输入数据D(N+1)使用系数矢量C2进行滤波
 - 则，C2的写入时序应当如何设置，才可以避免破坏D(N)滤波计算的正确性
 - 一种方案是使用乒乓系数RAM，仍需注意两片RAM的切换时序



算法设计—理论算法

- 原始公式
$$y(n) = \sum_{i=0}^{N-1} x(n-i)h(i)$$

- 并发计算公式

 - 例如：2个乘法累加单元

$$y_1(n) = \sum_{i=0}^{N/2-1} x(n-i)h(i), \quad y_2(n) = \sum_{i=N/2}^{N-1} x(n-i)h(i)$$

$$y(n) = y_1(n) + y_2(n)$$

 - N个单元的情况类似

算法设计—标号/地址映射

- 原始公式 $y(n) = \sum_{i=0}^{N-1} x(n-i)c(i)$
 - 为了避免数据RAM读写地址冲突，从最旧的数据开始读取冲突。令 $i = N-1-k$
 - 则 $y(n) = \sum_{i=0}^{N-1} x(n-(N-1-k)) \cdot c(N-1-k)$
 $y(n) = \sum_{k=0}^{N-1} x(n-N+1+k) \cdot c(N-1-k) \quad i, k = 0 \sim N-1$
 - k 为循环标号， x 和 c 的括号中的数作为RAM/ROM的地址
 - 地址序列是个有限域 $GF(N)$ ，则有 $k \pm N = k$
 - 对于系数ROM，读取地址为 $N-1-k$
 - 对于数据RAM
 - 读取地址为 $GF_N(n-N+1+k) = GF_N(n+1+k)$
 - 写入地址为 n

数据RAM读写地址举例

- 设有3个抽头系数
- 需保存3个历史数据
- 每次滤波计算，系数的读取顺序固定为 $c(2), c(1), c(0)$

$$\text{Read Addr} = (\text{Write Addr} + 1 + \text{Loop Index}) \text{ MOD } 3$$

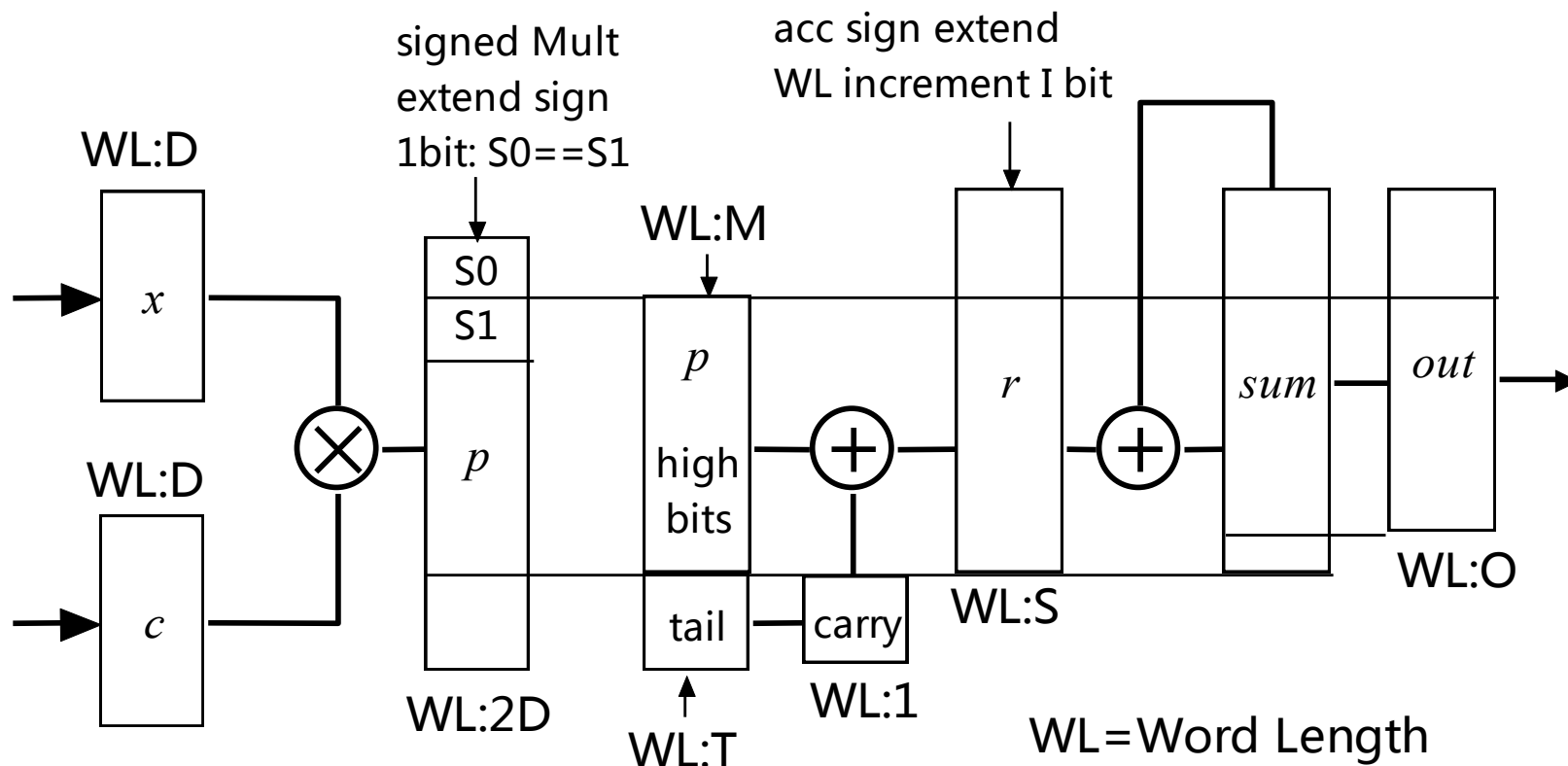
Input Sample	Loop Index →	0	1	2	Write Addr	Read Addr
	$x(0)$		$x(0)$	0	0	0
$x(1)$		$x(0)$	$x(1)$	0	1	2/0/1
$x(2)$		$x(0)$	$x(1)$	$x(2)$	2	0/1/2
$x(3)$		$x(3)$	$x(1)$	$x(2)$	0	1/2/0
$x(4)$		$x(3)$	$x(4)$	$x(2)$	1	2/0/1
$x(5)$		$x(3)$	$x(4)$	$x(5)$	2	0/1/2
$x(6)$		$x(6)$	$x(4)$	$x(5)$	0	1/2/0

data RAM content

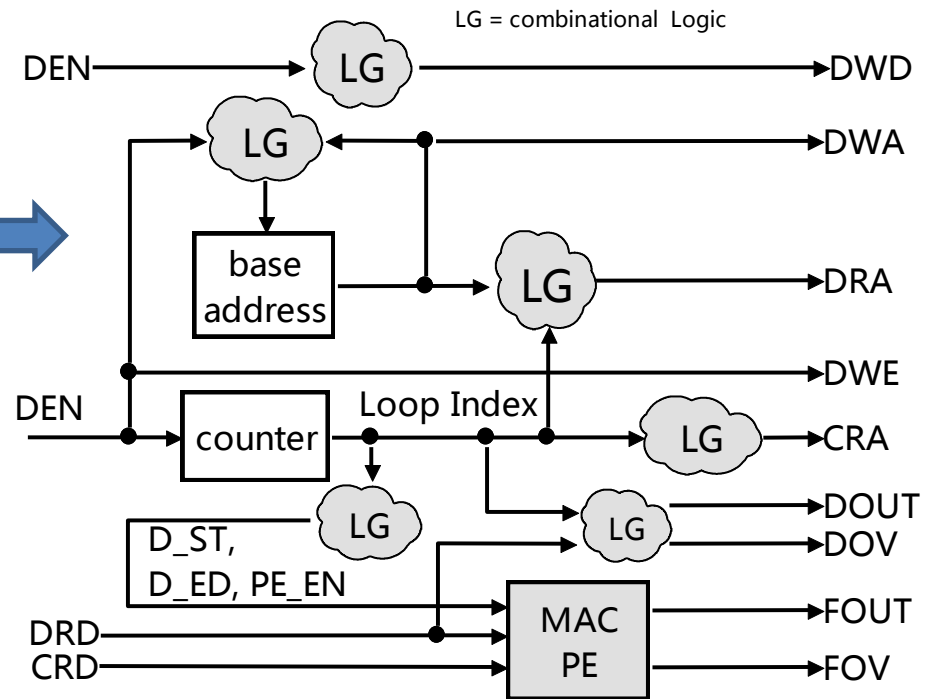
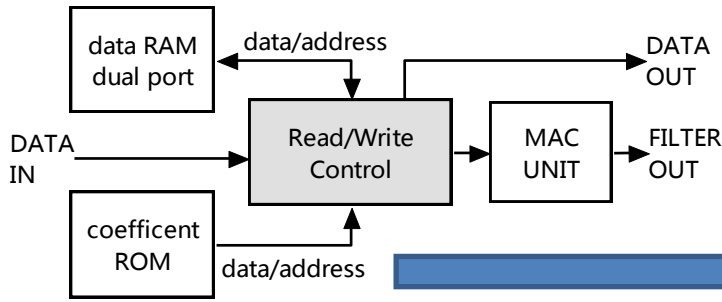
算法设计—定点策略

- 字长参数的依赖关系

- D/M/O 由用户设定。 $I = \lceil \log_2(N) \rceil$, $S = M + I$, $T = 2D - 1 - M$



电路结构——读写控制器模块结构



CRA: Coefficients Rom Read Address DEN: Data Input Enable D_ST :Data Sequence Start
 CRD: Coefficients Rom Read Data DIN: Data Input D_ED: Data Sequence End
 DRA: Data Ram Read Address FOUT: Filter Output PE:EN: MAC PE work enable
 DRD: Data Ram Read Data FOV: Filter Output Valid
 DWA: Data Ram Write Address DOUT: Data Output
 DWE: Data Ram Write Enable DOV: Data Output Valid
 DWD: Data Ram Write Data

- 右图为数据读写控制器电路的中间节点信号
- 其中省略了用于流水线对齐的D触发器
- 输入数据使能启动循环计数器
- 更新基地址寄存器
- 生成ROM读取地址
- 生成RAM的读写地址
- 标识出累加序列的起始D_ST和结尾D_ED

电路结构——读写控制器流水线对齐

Pipeline Stage

0	1	2	3
DIN(x(n)) DEN(x(n))	din_d1R den_d1R cra dra(x(n-N+1)) k_cnt_R n_ba_R(x(n)) dwe, dwa, dwd	din_d2R den_d2R dout_R, dov_R crd, drd(x(n-N+1)) d_st d_ed k_cnt_d1R dwd(done)	DOUT, DOV

CRA: Coefficients Rom Read Address DEN: Data Input Enable D_ST : Data Sequence Start
 DRA: Data Ram Read Address DIN: Data Input D_ED: Data Sequence End
 DRD: Data Ram Read Data FOUT: Filter Output PE:EN: MAC PE work enable
 DWA: Data Ram Write Address FOV: Filter Output Valid
 DWE: Data Ram Write Enable DOUT: Data Output
 DWD: Data Ram Write Data DOV: Data Output Valid

- 流水线的依据：
- 首先是必要流水：例如ROM的读数据和读地址之间，启动信号和D触发器之间：
- 然后是时序优化流水：例如为了提高fMAX，添加的D触发器

电路结构—读写控制器时序举例

- 设有3个抽头系数，需保存3个历史数据
- 每次滤波计算，系数的读取顺序固定为 $c(2), c(1), c(0)$ ，数据则从最旧的读到最新的
- 下图展示了数据的相对时间关系，以及数据RAM内容的变化，同种色调表示相同的一次滤波迭代。

<i>k</i>		0	1	2	0	1	2	0	1	2	
<i>DEN</i>	1	0	0	1	0	0	1	0	0		
<i>DIN</i>	$x(3)$	$x(3)$	$x(3)$	$x(4)$	$x(4)$	$x(4)$	$x(5)$	$x(5)$	$x(5)$		
<i>DWE</i>		1	0	0	1	0	0	1	0	0	
<i>DWA</i>		0	0	0	1	1	1	2	2	2	
<i>DWD</i>		$x(3)$	$x(3)$	$x(3)$	$x(4)$	$x(4)$	$x(4)$	$x(5)$	$x(5)$	$x(5)$	
<i>CRA</i>		2	1	0	2	1	0	2	1	0	
<i>CRD</i>			$c(2)$	$c(1)$	$c(0)$	$c(2)$	$c(1)$	$c(0)$	$c(2)$	$c(1)$	$c(0)$
<i>DRA</i>		1	2	0	2	0	1	0	1	2	
<i>DRD</i>			$x(1)$	$x(2)$	$x(3)$	$x(2)$	$x(3)$	$x(4)$	$x(3)$	$x(4)$	$x(5)$
<i>D_ST</i>			1	0	0	1	0	0	1	0	0
<i>D_ED</i>			0	0	1	0	0	1	0	0	1
0	$x(0)$	$x(0)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$	$x(3)$
1	$x(1)$	$x(1)$	$x(1)$	$x(1)$	$x(1)$	$x(4)$	$x(4)$	$x(4)$	$x(4)$	$x(4)$	$x(4)$
2	$x(2)$	$x(2)$	$x(2)$	$x(2)$	$x(2)$	$x(2)$	$x(2)$	$x(2)$	$x(5)$	$x(5)$	$x(5)$

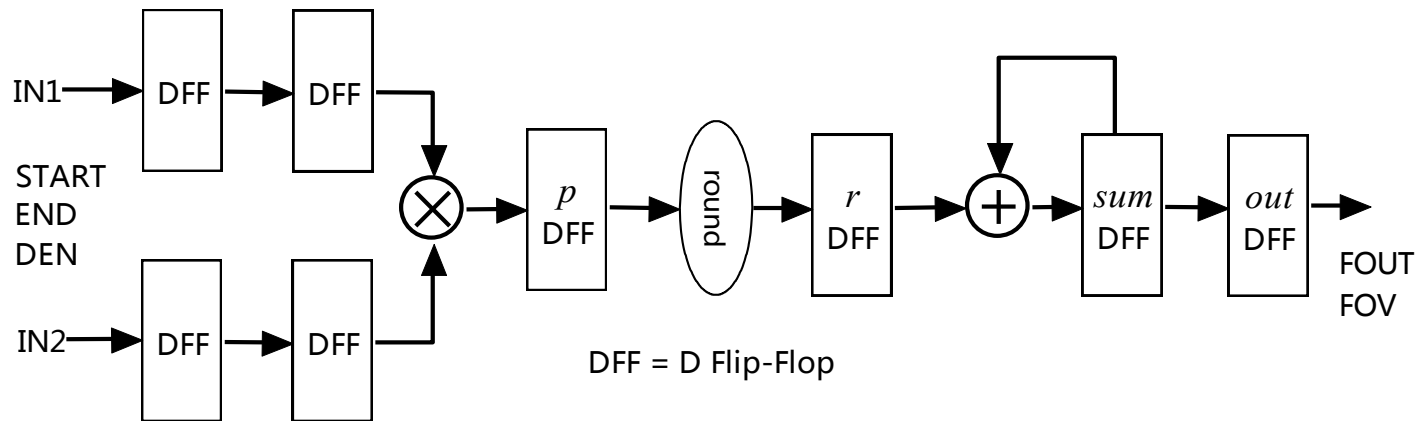
*RAM
ADDR*

RAM Content

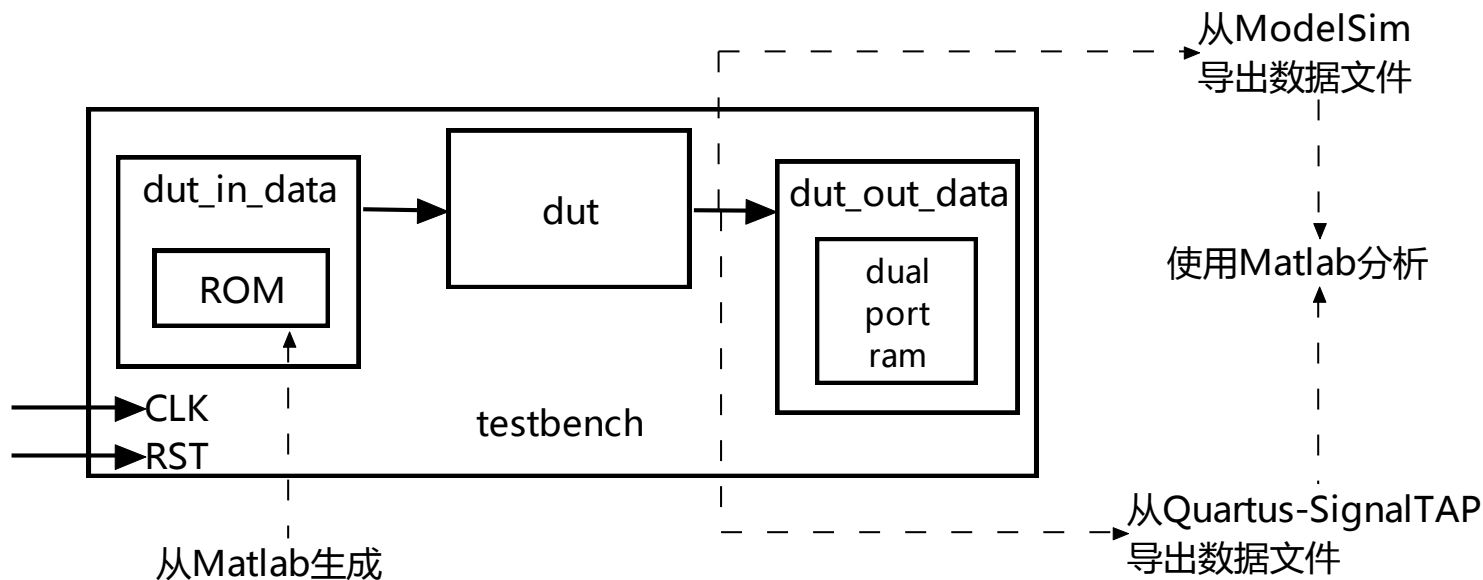
电路结构—乘法累加器

- 设计准则

- 每个运算单元的两端均用D触发器缓冲
- 使用DEN、START信号控制累加器的清零
- 使用DEN、END信号控制累加结果的输出
- 下图中省略了DEN、START、END信号的流水线对齐



设计验证



- 由Matlab生成激励数据ROM的HDL文件
- ModelSim进行仿真，导出数据文件（使用\$fwrite）
- 使用Quartus编译整个Testbench，使用SignalTAP抓取数据，导出List文件
- 详情请参考“可综合 RTL Testbench”参考设计

设计验证—Matlab配置

- 文件：gen_DUT_in_rom.m，生成激励信号数据和ROM的RTL代码文件
 - 激励信号的类型，多音正弦、白噪、chirp扫频
 - 激励信号的向量长度，采样率，量化位数
- gen_fir_coeff_rom.m，生成滤波器系数和ROM的RTL代码文件
 - 滤波器系数个数
 - 滤波器频响参数
 - 生成滤波器系数的matlab函数（fir2或firpm）
 - 滤波器系数的量化位数

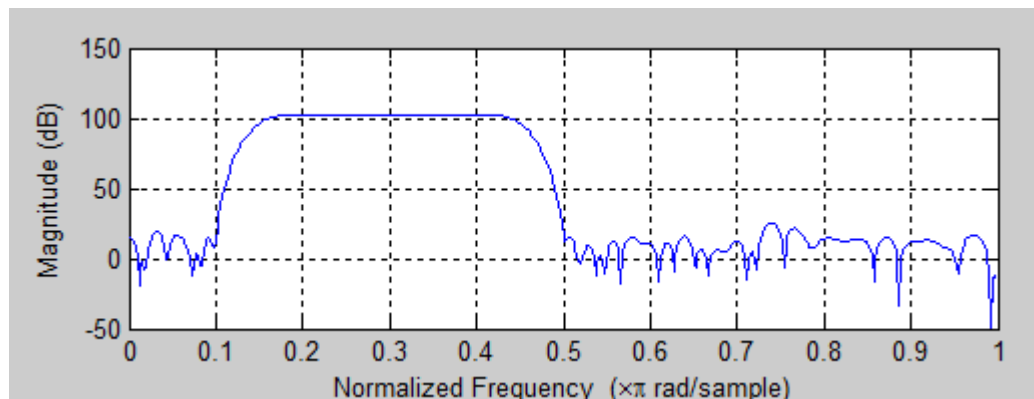
设计验证—Testbench和RTL配置

- 文件：testbench.v , module testbench()
 - 时钟周期
 - 激励信号样点数
 - 激励ROM地址宽度
 - 发送激励样点的时间间隔
 - 激励信号的向量长度，采样率，量化位数
 - RTL输入输出数据的字长
- 文件：testbench.v , module fir_inst()
 - 滤波器的输入字长
 - 滤波器的输出字长
 - 滤波器的ROM、RAM地址位宽
 - 滤波器系数个数

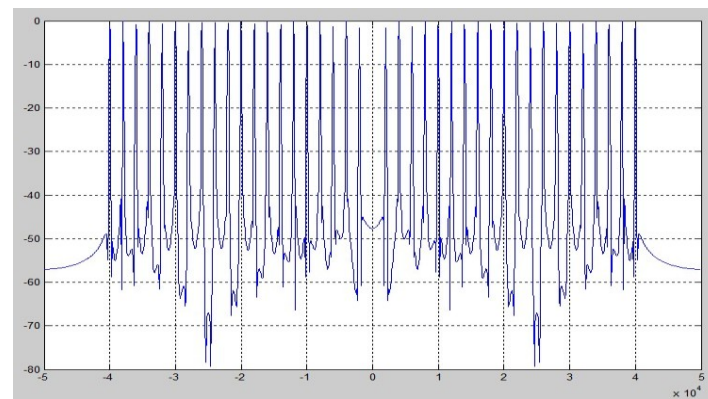
设计验证-ModelSim仿真

- 使用不同的信号进行测试
- 扫频信号，单音正弦，多音正弦，白噪声
- 观察时域波形或导出数据文件，用Matlab分析

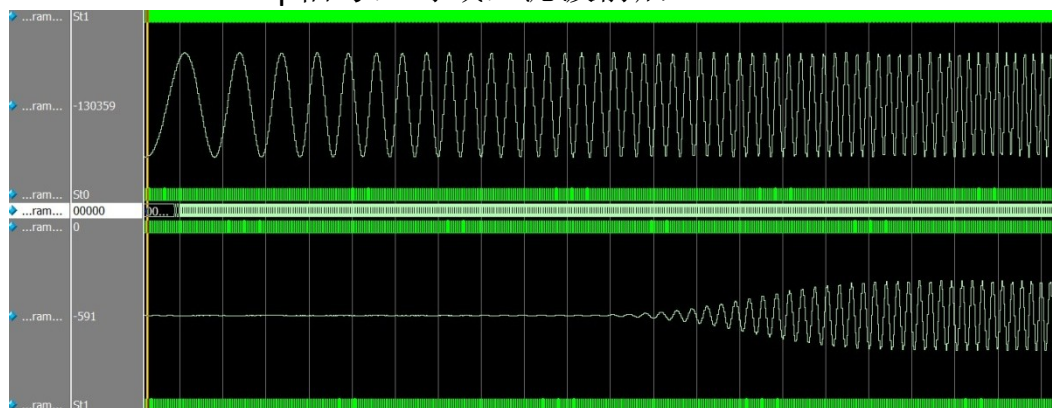
带通滤波器频响



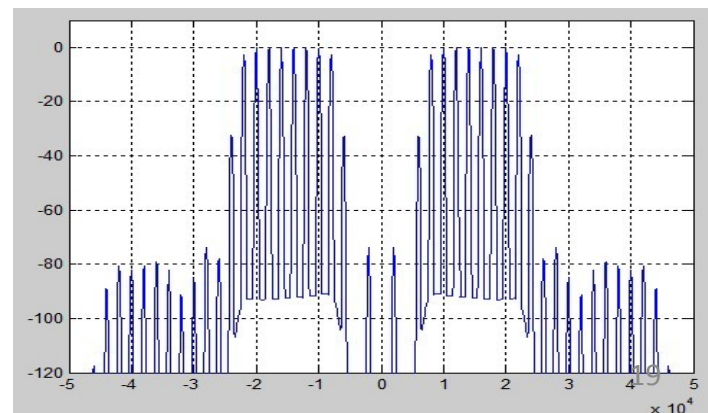
滤波器输入多音正弦频谱



Chirp信号，时域，滤波前后



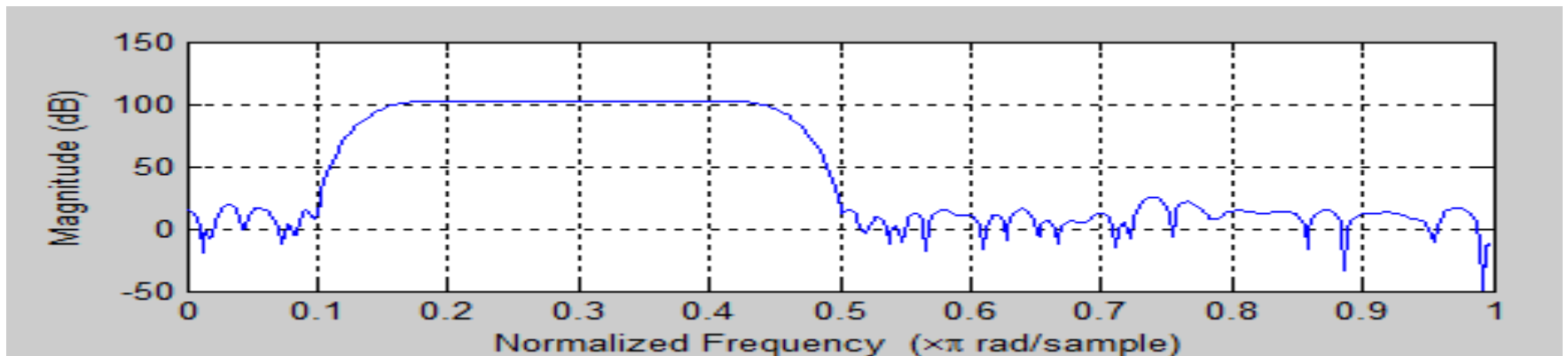
滤波器输出多音正弦频谱



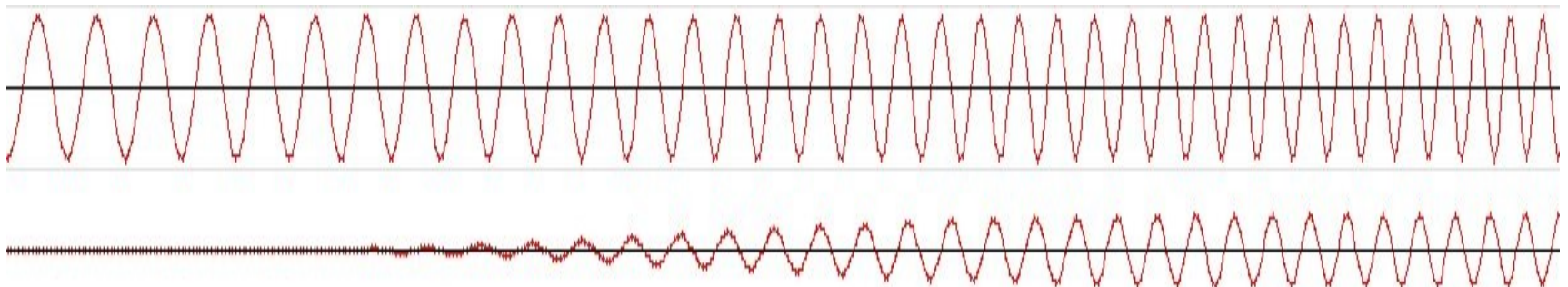
设计验证-SignalTAP导出数据

- 触发类型设定为segment，段长度为1
- 使用输出使能作为触发信号
- 观察时域波形或导出数据的list文件，用Matlab分析

带通滤波器频响



Chirp信号时域波形



*That's All
&
Thank You*