

# 中值滤波电路模块

## 设计文档

2022 年 07 月 01 日

小组成员：

1952033 侯雅玥

1952058 田海文

1952398 唐 超

1953143 秦善天

---

---

# 修订记录

版本	编辑注释	作者/修订时间	审核/审核时间	
0.1	初始文档	xxx/20220512		
0.2	文档更新	xxx/20220701		

---

# 目录

<b>1. 模块概要 .....</b>	<b>1</b>
1.1. 功能介绍 .....	1
1.2. 模块特征定义 .....	1
1.3. 文档命名规则 .....	1
<b>2. 顶层概要 .....</b>	<b>3</b>
2.1. 模块框图 .....	3
2.2. 端口信号一览 .....	3
2.3. 接口时序和数据格式 .....	3
2.4. 逻辑设计层次和文件列表 .....	5
2.5. 模块内部数据流程图 .....	6
2.6. 模块中的内存和数据格式 .....	6
2.7. 模块中使用到的算法 .....	6
2.8. 模块中时钟域 .....	8
2.9. 模块的复位方法 .....	8
<b>3. 软件相关 .....</b>	<b>9</b>
3.1. 寄存器列表 .....	9
3.2. 配置流程和仿真结果 .....	10
<b>4. 集成介绍 .....</b>	<b>11</b>
4.1. 系统集成框图 .....	11
4.2. 时钟要求 .....	11
4.3. 复位要求 .....	11
4.4. 综合约束(synthesis constraint) .....	11
4.5. 综合报告（将 quartus 综合之后的资源使用情况和最大时钟频率这些内容放在本部分） .....	11
<b>5. 参考文档 .....</b>	<b>15</b>

---

# 图目录

Figure 2-1 模块框图 .....	3
Figure 2-2 APB2 写操作时序 .....	4
Figure 2-3 APB2 读操作时序 .....	4
Figure 2-4 SRAM 写操作时序图 .....	5
Figure 2-5 SRAM 读操作时序图 .....	5
Figure 2-6 代码层级图 .....	5
Figure 2-7 模块内部数据流程图.....	6
Figure 2-8 SRAM 存储空间.....	6
Figure 2-9 九选一中值算法原理图 .....	7
Figure 2-10 九选一中值仿真波形图 .....	7
Figure 2-11 中值滤波模块状态机示意图.....	8
Figure 3-1 仿真结果波形图 .....	10
Figure 4-1 在系统集成中本模块所处位置 .....	11
Figure 4-2 中值滤波模块综合原理图 .....	12

---

## 1. 模块概要

### 1.1. 功能介绍

本模块主要完成数字图像处理的中值滤波算法，滤波窗口为 3x3。

模块经过软件配置后，可以对存放在外部 memory 输入区域中的灰度图像进行中值滤波，滤波之后的图像保存在外部 memory 的输出区域；

图像滤波完成之后，产生中断信号。

### 1.2. 模块特征定义

- 最大支持 8192x8192 的图像
- 支持 120fps@1280x1024
- 采用 32 位宽的 APB 2.0 总线进行系统配置
- 提供灵活的参数化配置机制
- 模块工作时钟的最大时钟频率 200MHz

### 1.3. 文档命名规则

信号中的比特位用尖括号标示

EXTERNAL\_ADDRESS<31:0>

nCS<1>

寄存器中的比特位用方括号标示

REGISTER\_BITFIELD[3:0]

REGISTER\_BIT[0]

下列行为用专用词汇描述，减少大家的混淆

**clear**—对单个寄存器比特编程为 0（Program 1b'0 into a single register bit）

**set**—对单个寄存器比特编程为 1（Program 1b'1 into a single register bit）

**write**—对多个集群其比特位编程（Program a hexadecimal value into a register bit field (more than one bit)）

**assert**—驱动信号到有效电平（Drive a signal to its active voltage level, either high or low）

**de-assert**—驱动信号到无效电平（Drive a signal to its inactive voltage level, either high or low.）

**drive**—驱动信号到高电平（Assert a voltage level onto a signal.）

寄存器定义表格的描述，应遵循以下规则

‘Reset’ 列中的值应该符合以下描述

0 = Bit clear

---

**1** = Bit set

‘Access’ 列中的值应该符合以下描述

**RO** = Read-only

**WO** = Write-only

**R/W** = Read and write

There are two special cases:

**R/WC** = R/W. To clear the bit, write 1'b1 to it.

**RC** = Read-only. The bit is automatically cleared after it is read.

## 2. 顶层概要

### 2.1. 模块框图

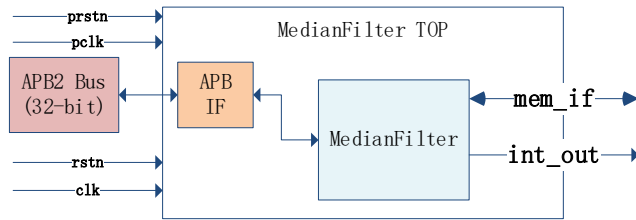


Figure 2-1 模块框图

### 2.2. 端口信号一览

Table 2-1

信号名	信号宽度	信号方向	信号缺省值 (only output for)	信号说明
pclk	<0:0>	输入		APB 时钟
prstn	<0:0>	输入		APB 时钟域复位信号，低电平有效
clk	<0:0>	输入		模块工作时钟
rstn	<0:0>	输入		模块工作时钟域复位信号，低电平有效
APB 2 Bus				APB 2 Slave, 32 位总线信号
int_out	<0:0>	输出		模块中断输出信号，高电平有效
mem_csn	<0:0>	输出		Memory chip select, 低有效
mem_wen	<7:0>	输出		Memory write enable, 低电平为写操作，所有比特均为高电平则进行读操作
mem_adr	<20:0>	输出		地址信号输出到 Memory
mem_do	<63:0>	输出		数据信号输出到 Memory
mem_di	<63:0>	输入		来自 Memory 的数据信号输入

### 2.3. 接口时序和数据格式

接口包括 APB 2，其接口时序均符合标准。  
APB2 写操作的时序图如下：

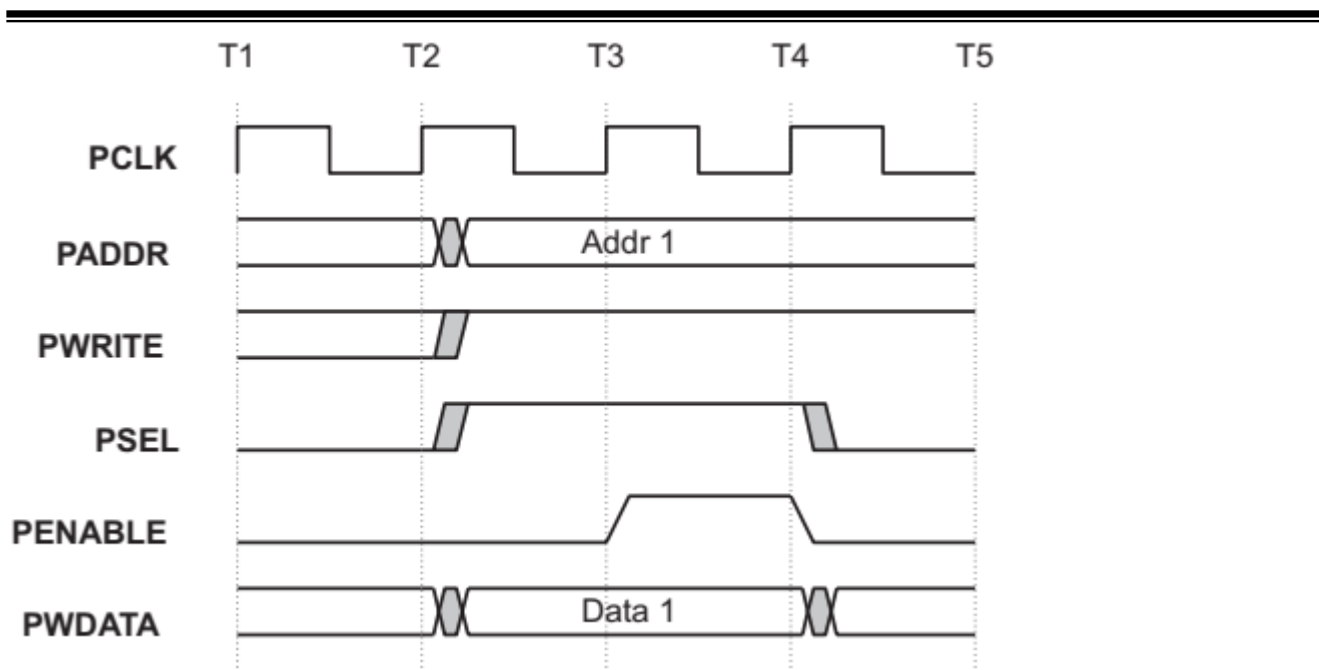


Figure 2-2 APB2 写操作时序

APB2 读操作的时序图如下：

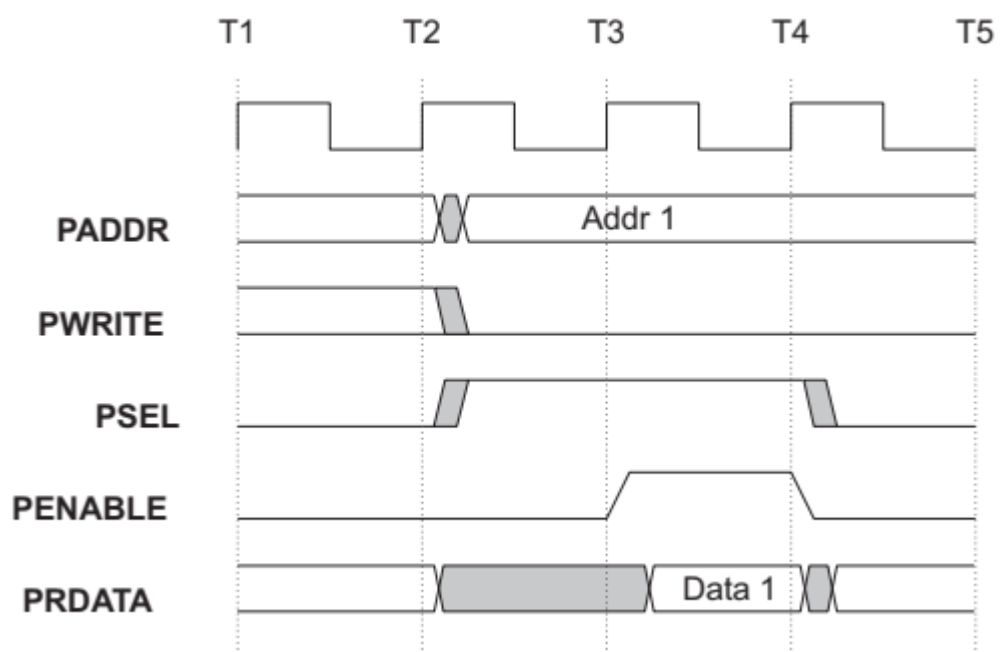


Figure 2-3 APB2 读操作时序

SRAM 写操作的时序图如下：



### 写SRAM

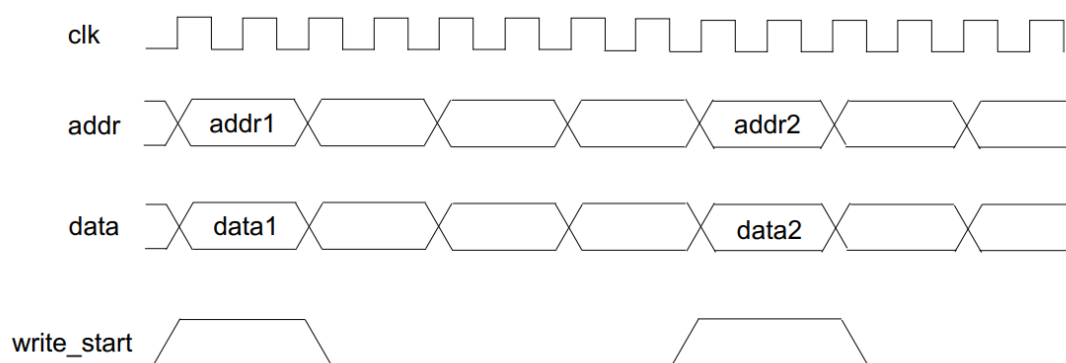


Figure 2-4 SRAM 写操作时序图

SRAM 读操作的时序图如下：

### 读SRAM

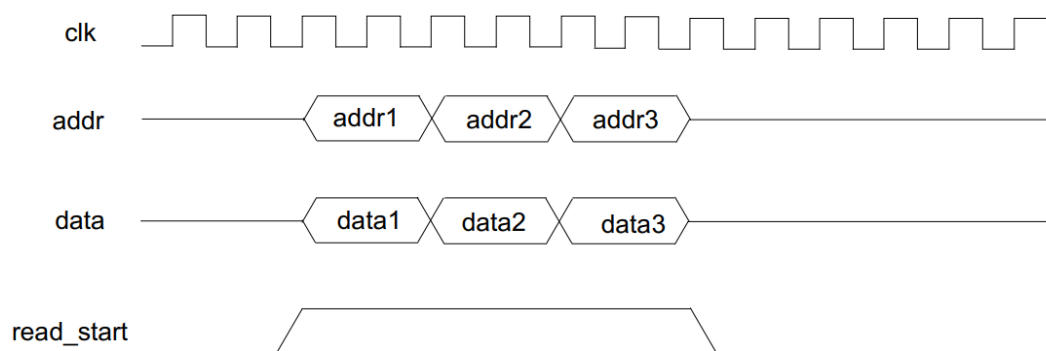


Figure 2-5 SRAM 读操作时序图

## 2.4. 逻辑设计层次和文件列表

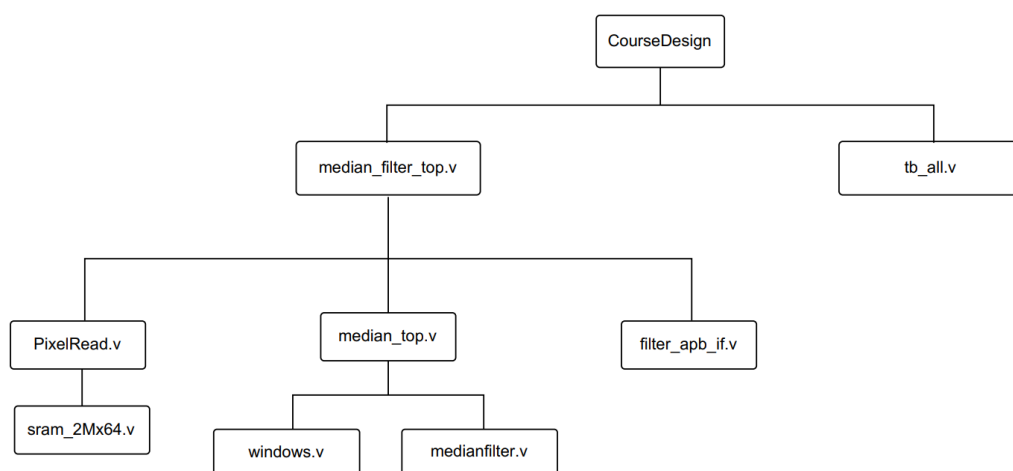


Figure 2-6 代码层级图

## 2.5. 模块内部数据流程图

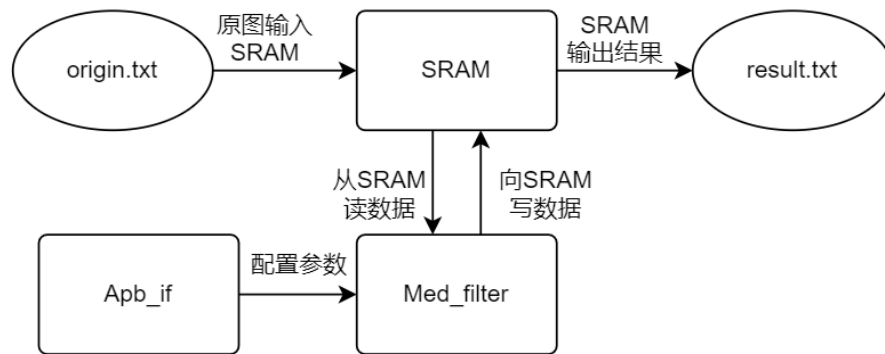


Figure 2-7 模块内部数据流程图

## 2.6. 模块中的内存和数据格式

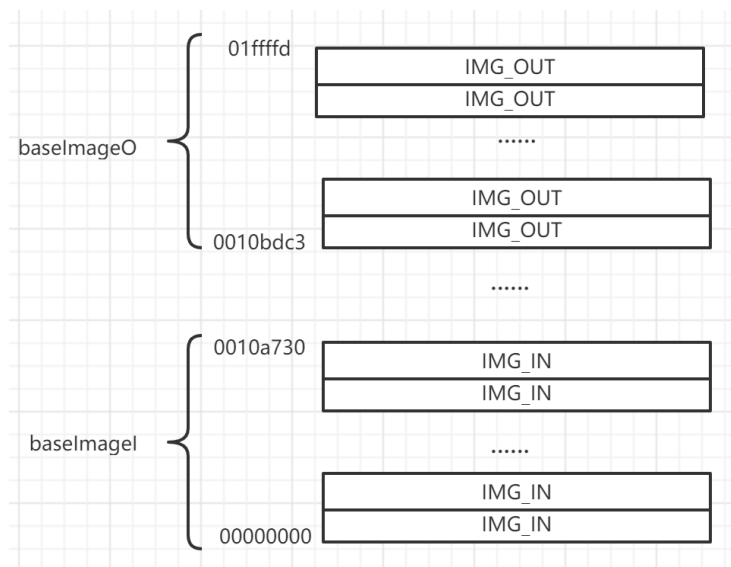


Figure 2-8 SRAM 存储空间

SRAM 模块为 64bitx2M 大小的内存块。内存存储结构为每一个 21 位宽的地址指向一块 64bit 的空间，每一个地址指向的空间中存储 4 个大小为 16bit 的像素，像素按图像位置顺序排列。

SRAM 内存中的内存分为两部分，其中 0-0x0010a730 为原始图像存储单元，0x0010bdc3-0x01ffffd 为处理后数据存储单元。将原始图像数据转换为.txt 文件后，按原始图像的像素顺序存储进 SRAM。

clk 为时钟信号，csn 为读写操作使能信号，低电平有效，adr 为地址总线，din 为输入数据线，dout 为输出数据线。

## 2.7. 模块中使用到的算法

### 2.7.1 九选一中值算法

实现流程：

1) 并行输入 9 个数（3 行 3 列），利用比较器进行排序，分别取得每一行数据的最大值、中间值和最小值；

- 2) 第二次排序，3 个最大值，3 个最小值，3 个中间值分别比较，取 3 个最大值的最小值，3 个最小值的最大值，3 个中间值的中间值；
- 3) 将上述三个数据排序，得到的中间值即为九个数的中间值。

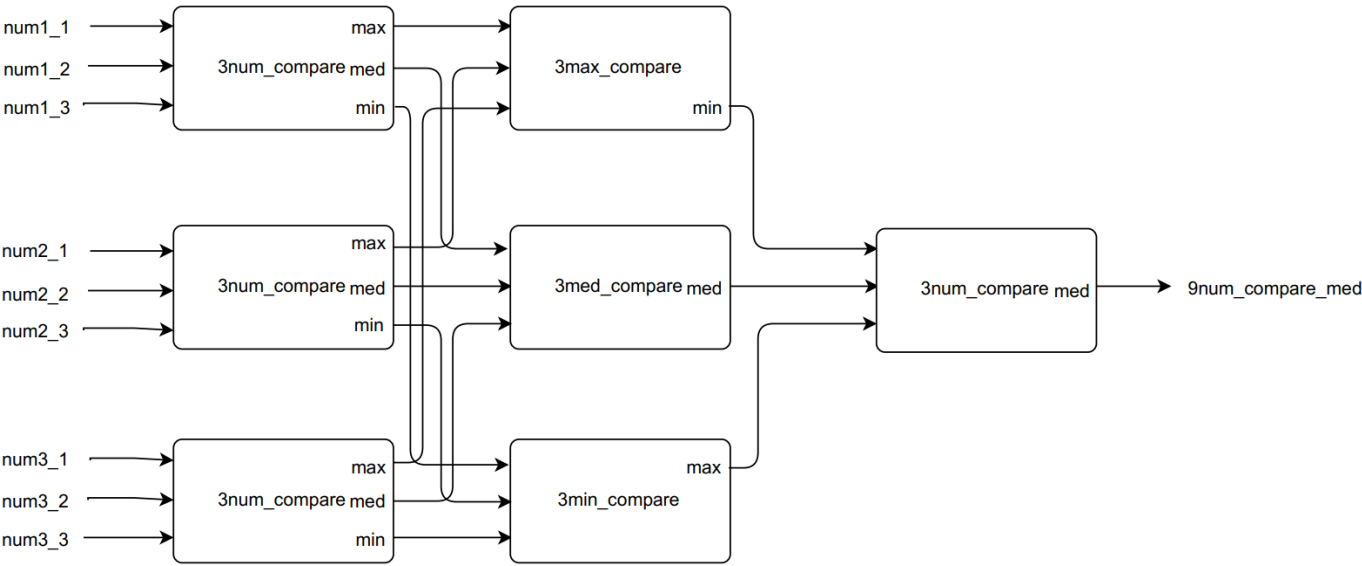


Figure 2-9 九选一中值算法原理图

功能仿真：

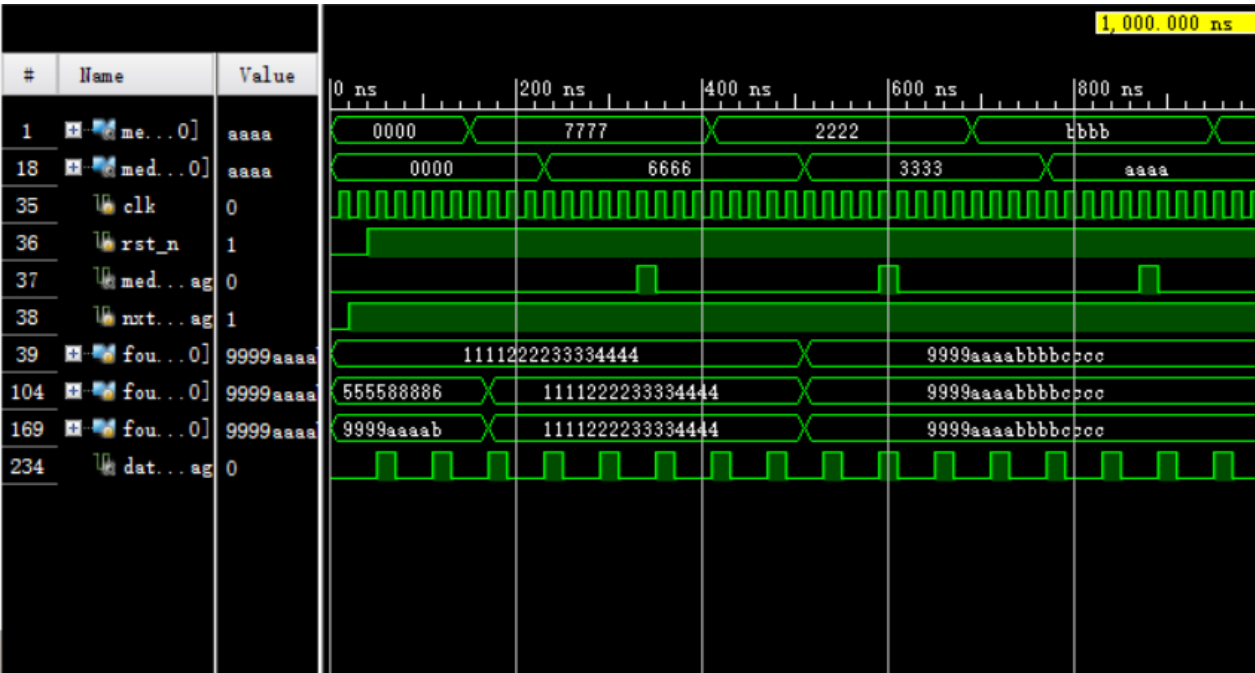


Figure 2-10 九选一中值算法仿真波形图

### 2.7.2 中值滤波模块状态机

中值滤波模块工作时主要分成 5 个状态，其状态机示意图如下所示：

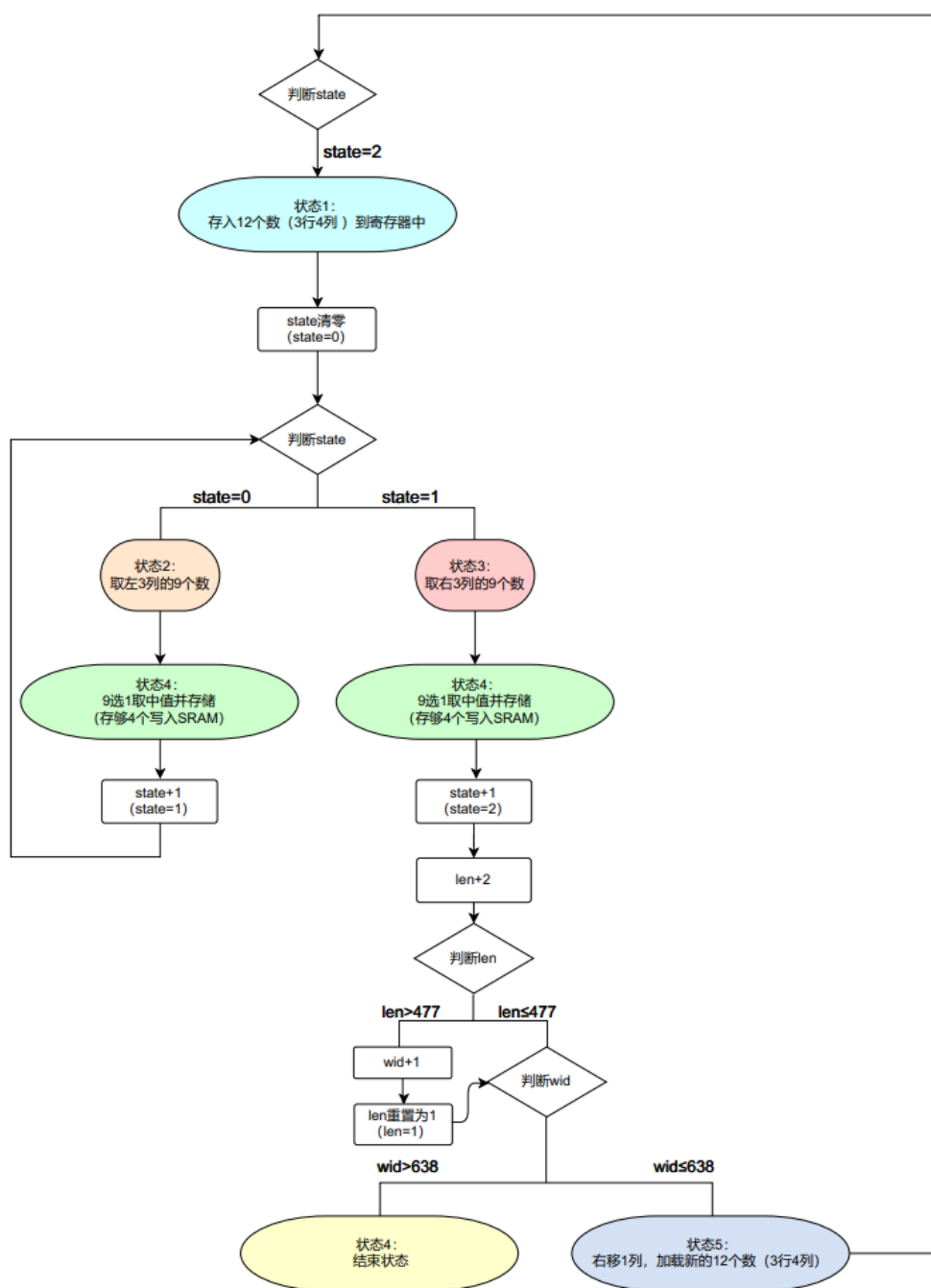


Figure 2-11 中值滤波模块状态机示意图

## 2.8. 模块中时钟域

APB 接口模块采用 APB 时钟，时钟信号来自于芯片的 clock generator。  
本模块的工作时钟信号为 clk，来自于芯片的 clock generator。

## 2.9. 模块的复位方法

本模块的所有子模块的复位信号（presetn 和 rstn）来自于芯片级的 reset generator。

### 3. 软件相关

具体软件配置流程如下：

#### 3.1. 寄存器列表

Table 3-1

偏移量	寄存器名	存取方式	默认值	寄存器说明
16'h0000	Control	读写	0	控制寄存器
16'h0004	Frame_width	读写	1280	图像宽度
16'h0008	Frame_height	读写	1024	图像高度
16'h000c	Input_base_address	读写	0	输入图像在外部 memory 中的起始地址，以 byte 为单位
16'h0010	Output_base_address	读写	0	输出图像在外部 memory 中的起始地址，以 byte 为单位
16'h0014	Frame_number	只读	0	已滤波图像的总帧数
16'h0018	Cur_Frame_Cycle	只读	0	当前帧图像滤波所用时钟数
16'h001c	Total_Frame_Cycle	只读	0	所有帧图像滤波所用时钟数
16'h0020	Int status	读写	0	中断源
16'h0024	Int enable	读写	0	中断使能

##### 3.1.1 CONTROL 寄存器

Bit[0]=1: start filter, 0: none

Bit[1]=1: 输入图像每个像素为两字节, 0: 输入图像每个像素为单字节

Bit[31:2]: reserved ;

##### 3.1.2 Frame\_width 寄存器

Bit[15:0]=Frame\_width, 输入图像的水平尺寸，表示图像一行像素总数为 Frame\_width;

Bit[31:16]: reserved ;

##### 3.1.3 Frame\_height 寄存器

Bit[15:0]=Frame\_height, 输入图像的垂直尺寸，表示图像总行数为 Frame\_height;

Bit[31:16]: reserved ;

##### 3.1.4 Input\_base\_address 寄存器

Bit[23:0]= Input\_base\_address, 输入图像在外部 memory 中的起始地址，以 byte 为单位;

Bit[31:24]: reserved ;

##### 3.1.5 Output\_base\_address 寄存器

Bit[23:0]= Output\_base\_address, 输出图像在外部 memory 中的起始地址，以 byte 为单位;

Bit[31:24]: reserved ;

##### 3.1.6 Frame\_number 寄存器

Bit[31:0]= Frame\_number, 已滤波图像的总帧数;

##### 3.1.7 Cur\_Frame\_Cycle 寄存器

Bit[31:0]= Cur\_Frame\_Cycle, 当前帧图像滤波所用时钟数;

---

### 3.1.8 Total\_Frame\_Cycle 寄存器

Bit[31:0]= Total\_Frame\_Cycle: 所有帧图像滤波所用时钟数;

### 3.1.9 Int Status 寄存器

Bit[0]: filter frame done

Bit[31:1]: reserved ;

### 3.1.10 Int enable 寄存器

Bit[0]: interrupt enable for filter frame done

Bit[31:1]: reserved ;

## 3.2. 配置流程和仿真结果

### 1. 软件配置流程:

- 1) 配置时钟激励源, 20ns 为一个时钟周期, 并将系统内全部时钟都统一为一个时钟。
- 2) 配置 apb 总线, 写入存储初始地址, 图片尺寸、图片帧数等信息。
- 3) 配置复位信号与使能信号。
- 4) 将原始文件通过\$fsconf 逐行存入 SRAM。
- 5) 例化顶层模块, 进行处理。
- 6) 结束仿真。

### 2. 仿真结果:

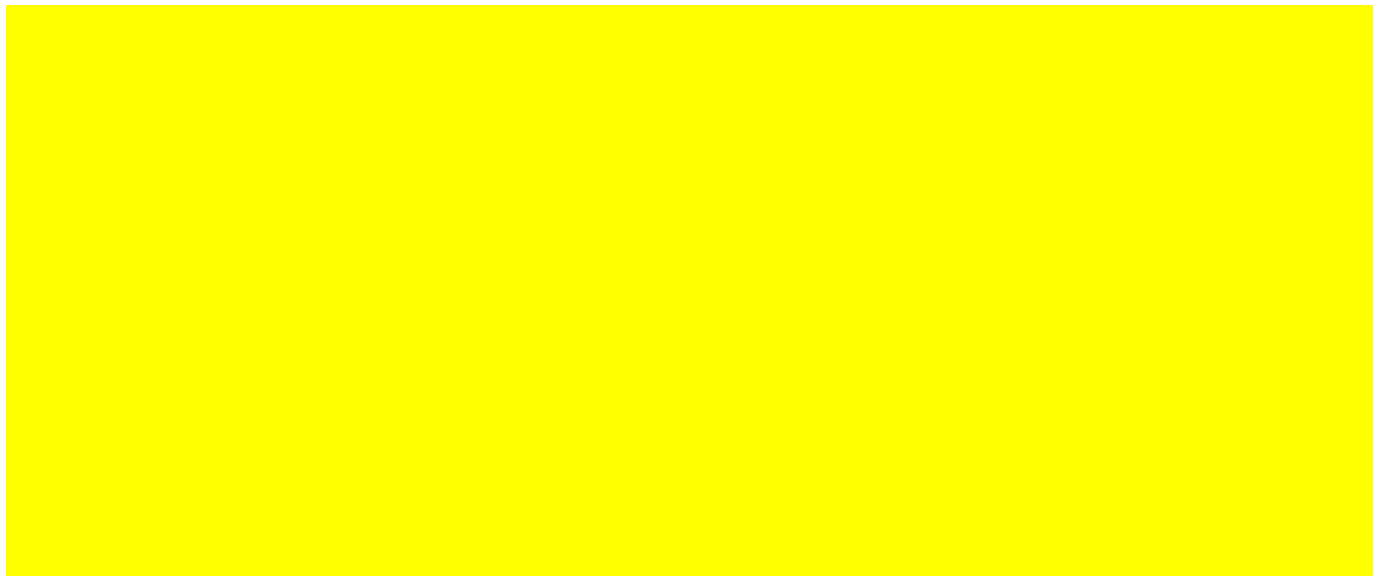


Figure 3-2 仿真结果波形图

---

## 4. 集成介绍

### 4.1. 系统集成框图

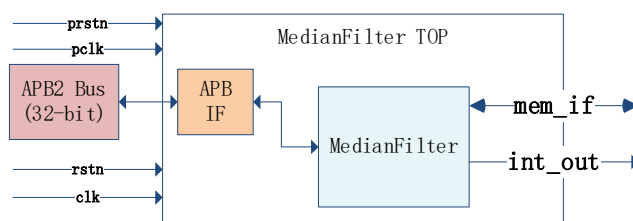


Figure 4-1 在系统集成中本模块所处位置

如图 4-1，本模块通过 APB 2 接口进行参数配置。

### 4.2. 时钟要求

本模块的有 2 个时钟输入，全部来自片内的 Clock Generator。

此 2 个时钟之间均为异步时钟域。

### 4.3. 复位要求

本模块中，所有的复位信号都是同步复位，各个时钟域均有相应的复位信号进入本模块顶层。芯片级的 Reset Generator 应将各个复位信号同步后再送到本模块。

### 4.4. 综合约束(synthesis constraint)

#### 1. False Path List:

set\_false\_path -from pclk -to clk

set\_false\_path -from clk -to pclk

2. 本模块并无 MCP 路径。

3. 建议 input/output delay 均设为相应时钟周期的 30%

### 4.5. 综合报告

#### 1. 中值滤波模块综合报告:

##### 1) Design Timing Summary:





8 Input	64 Bit	Muxes := 6
2 Input	32 Bit	Muxes := 1
2 Input	21 Bit	Muxes := 3
8 Input	21 Bit	Muxes := 3
2 Input	18 Bit	Muxes := 2
8 Input	18 Bit	Muxes := 2
2 Input	1 Bit	Muxes := 2
8 Input	1 Bit	Muxes := 2
Module windows		
Detailed RTL Component Info :		
+---Registers :		
	64 Bit	Registers := 3
	16 Bit	Registers := 9
	1 Bit	Registers := 2
+---Muxes :		
2 Input	16 Bit	Muxes := 9
3 Input	3 Bit	Muxes := 1
2 Input	1 Bit	Muxes := 1
3 Input	1 Bit	Muxes := 3
Module medianfilter		
Detailed RTL Component Info :		
+---Adders :		
2 Input	3 Bit	Adders := 1
+---Registers :		
	16 Bit	Registers := 23
	3 Bit	Registers := 1
	2 Bit	Registers := 1
	1 Bit	Registers := 2
+---Muxes :		
2 Input	16 Bit	Muxes := 49
2 Input	3 Bit	Muxes := 1
2 Input	2 Bit	Muxes := 2
3 Input	1 Bit	Muxes := 1
2 Input	1 Bit	Muxes := 11
4 Input	1 Bit	Muxes := 5
6 Input	1 Bit	Muxes := 1
7 Input	1 Bit	Muxes := 1
Module filter_apb_if		
Detailed RTL Component Info :		
+---Registers :		
	32 Bit	Registers := 3
	16 Bit	Registers := 2
	2 Bit	Registers := 1
	1 Bit	Registers := 2
+---Muxes :		
2 Input	1 Bit	Muxes := 1

---

---

Finished RTL Hierarchical Component Statistics

---

## 5. 参考文档

1. APB\_2.0 Specification.