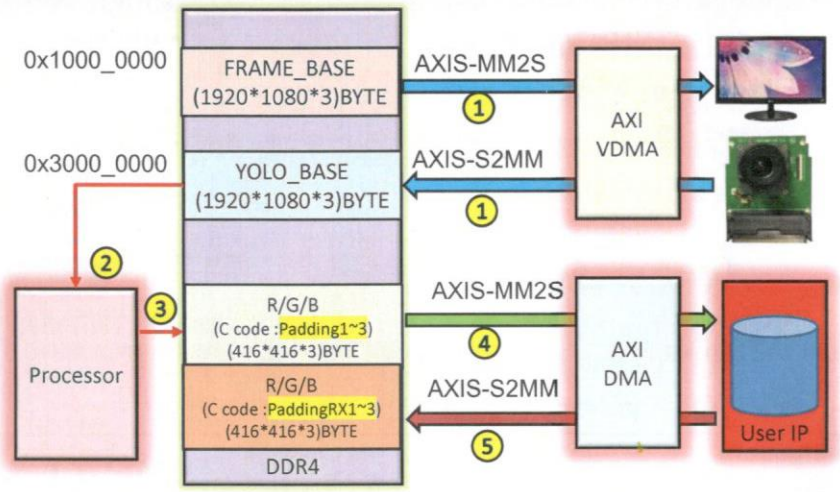
isif_data_out	in	64	Data from input stream FIFO

osif_data_din	out	64	Data output to stream FIFO

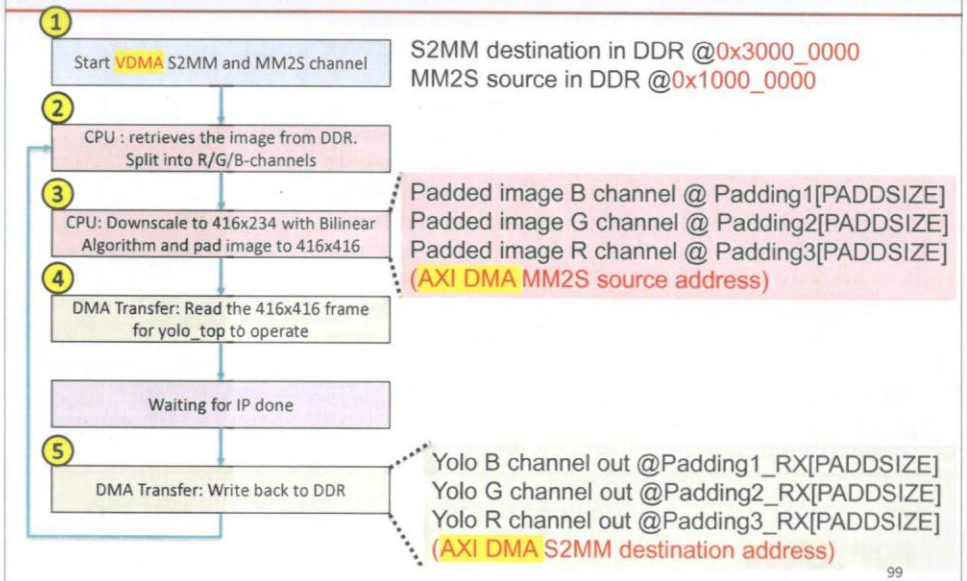
NARLabs

Video Data Flow



NARLabs

FW Design Flow



NARLabs

main.c Code Start

main.c | gpio.c | draw.c | pipeline_pro... | sleep.c

```
////// Code start

        //AXI_DMA initial add++++++++++++++++++
        AXI_DMA_Init();
        //-----------------------------------

        //AXI GPIO Test++++++++++++++++++++++++
        gpio_test();
        //-----------------------------------
        while(1){
            //usleep(85000);//simulation wait IP to ready
#if 1
            //Keep the first frame
            memcpy(FstFrame, (unchar *)DRAW_BASE, TBSIZE);

            //Split R/G/B channel
            for(int iCnt = 0; iCnt < CHNLSIZE; iCnt++)
            {
                chn1[iCnt] = FstFrame[3*iCnt];      //???B
                chn2[iCnt] = FstFrame[3*iCnt + 1];  //???G
                chn3[iCnt] = FstFrame[3*iCnt + 2];  //???R
            }
```

Initial AXI DMA

GPIO TEST

Copy (0x1000_0000 1920*1080 byte)to FstFrame(array) ②

Split R/G/B channel ③

NARLabs

main.c Code

main.c | gpio.c | draw.c | pipeline_pro... | sleep.c | axi_dma.c | system.hdf

```
//AXI_DMA_IP_TEST();
#endif
#if 1
        AXI_DMA_TxDone=0;
        AXI_DMA_RxDone=0;
        AXI_DMA_Transfer((UINTPTR)(&Padding1)   ,PADDSIZE,  XAXIDMA_DMA_TO_DEVICE);
        AXI_DMA_Transfer((UINTPTR)(&Padding1_RX),PADDSIZE,  XAXIDMA_DEVICE_TO_DMA);
#if 0
        Xil_DCacheFlushRange((UINTPTR)Padding1, PADDSIZE);
        Xil_DCacheFlushRange((UINTPTR)Padding1_RX, PADDSIZE);
#endif
        //for verify data
        for(int i=0;i<PADDSIZE;i++)
        {
            if(Padding1_RX[i]!=Padding1[i])
            {
                xil_printf("Padding1 error \r\n");
            }
        }
        while (!AXI_DMA_TxDone && !AXI_DMA_RxDone ) {}
        AXI_DMA_TxDone=0;
        AXI_DMA_RxDone=0;

        AXI_DMA_Transfer((UINTPTR)(&Padding2)   ,PADDSIZE,  XAXIDMA_DMA_TO_DEVICE);
        AXI_DMA_Transfer((UINTPTR)(&Padding2_RX),PADDSIZE,  XAXIDMA_DEVICE_TO_DMA);
#if 0
        Xil_DCacheFlushRange((UINTPTR)(&Padding2), PADDSIZE);
        Xil_DCacheFlushRange((UINTPTR)(&Padding2_RX), PADDSIZE);
#endif
        //for verify data
        for(int i=0;i<PADDSIZE;i++)
        {
            if(Padding2_RX[i]!=Padding2[i])
            {
                xil_printf("Padding2 error \r\n");
            }
        }
        while (!AXI_DMA_TxDone && !AXI_DMA_RxDone ) {}
        AXI_DMA_TxDone=0;
        AXI_DMA_RxDone=0;
```

B Byte

G

Transfer data from DDR to IP ④

Transfer data from IP to DDR ⑤

Wait AXI DMA transfer complete

Transfer data from DDR to IP ④

Transfer data from IP to DDR ⑤

Wait AXI DMA transfer complete