Hello dear engineer, I have some questions about the EDMA3 technology of C6678.

I wrote a simple case based on the data transposition section of User Manual 3.3, where EDMA3 uses the CSL library.

The code is as follows:

**#include** <stdio.h>

**#include** <ti/csl/csl\_edma3.h>

**#include** <ti/csl/csl\_edma3Aux.h>

**#include** <ti/csl/csl\_cacheAux.h>

**#include** <stdlib.h>

Int32 **Data\_Sorting** (Int32 instNum, Uint8 channelNum,**float** \* srcBuff,**float** \*dstBuff,Int32 aCnt,

 Int32 bCnt,Int32 cCnt,Int32 srcBidx,Int32 dstBidx,Int32 srcCidx,Int32 dstCidx)

{

 CSL\_Edma3Handle hModule;

 CSL\_Edma3Obj edmaObj;

 CSL\_Edma3ParamHandle htrans;

 CSL\_Edma3ChannelObj chObj;

 CSL\_Edma3CmdIntr regionIntr;

 CSL\_Edma3ChannelHandle hChannel;

 CSL\_Edma3ParamSetup myParamSetup;

 CSL\_Edma3Context context;

 CSL\_Edma3ChannelAttr chAttr;

 CSL\_Status status;

 /\* Module initialization \*/

 **CSL\_edma3Init**(&context) != CSL\_SOK;

 /\* Open the EDMA Module using the provided instance number \*/

 hModule = **CSL\_edma3Open**(&edmaObj, instNum, NULL, &status);

 /\* Channel open \*/

 chAttr.regionNum = CSL\_EDMA3\_REGION\_GLOBAL;

 chAttr.chaNum = channelNum;

 hChannel = **CSL\_edma3ChannelOpen**(&chObj, instNum, &chAttr, &status);

 /\* For EDMA instance 1 and 2 maximum of 4 TCs and 4 event queues are supported

 \* Change Channel Default queue setup from 0 to 3

 \*/

 **CSL\_edma3HwChannelSetupQue**(hChannel,*CSL\_EDMA3\_QUE\_3*) ;

 /\* Map the DMA Channel to PARAM Block 2. \*/

 CSL\_edma3MapDMAChannelToParamBlock (hModule, channelNum, 1);

 /\* Obtain a handle to parameter set 2 \*/

 htrans = **CSL\_edma3GetParamHandle**(hChannel, 1, &status);

 myParamSetup.option = CSL\_EDMA3\_OPT\_MAKE(CSL\_EDMA3\_ITCCH\_EN, \

 CSL\_EDMA3\_TCCH\_DIS, \

 CSL\_EDMA3\_ITCINT\_DIS, \

 CSL\_EDMA3\_TCINT\_EN, \

 0, CSL\_EDMA3\_TCC\_NORMAL,\

 CSL\_EDMA3\_FIFOWIDTH\_NONE, \

 CSL\_EDMA3\_STATIC\_DIS, \

 CSL\_EDMA3\_SYNC\_AB, \

 CSL\_EDMA3\_ADDRMODE\_INCR, \

 CSL\_EDMA3\_ADDRMODE\_INCR );

 myParamSetup.srcAddr = (Uint32)srcBuff;

 myParamSetup.aCntbCnt = CSL\_EDMA3\_CNT\_MAKE(aCnt,bCnt);

 myParamSetup.dstAddr = (Uint32)dstBuff;

 myParamSetup.srcDstBidx = CSL\_EDMA3\_BIDX\_MAKE(srcBidx,dstBidx);

 myParamSetup.linkBcntrld= CSL\_EDMA3\_LINKBCNTRLD\_MAKE(0xffff,0);

 myParamSetup.srcDstCidx = CSL\_EDMA3\_CIDX\_MAKE(srcCidx,dstCidx);

 myParamSetup.cCnt = cCnt;

 /\* Ping setup \*/

 **CSL\_edma3ParamSetup**(htrans,&myParamSetup);

 /\* Interrupt enable (Bits 0-1) for the global region interrupts \*/

 regionIntr.region = CSL\_EDMA3\_REGION\_GLOBAL;

 regionIntr.intr = 0x3;

 regionIntr.intrh = 0x0000;

 **CSL\_edma3HwControl**(hModule,*CSL\_EDMA3\_CMD\_INTR\_ENABLE*,&regionIntr);

 /\* Trigger channel \*/

 **CSL\_edma3HwChannelControl**(hChannel,*CSL\_EDMA3\_CMD\_CHANNEL\_SET*,NULL);

 regionIntr.region = CSL\_EDMA3\_REGION\_GLOBAL;

 regionIntr.intr = 0;

 regionIntr.intrh = 0;

 /\* Poll on IPR bit 0 \*/

 **do** {

 **CSL\_edma3GetHwStatus**(hModule,*CSL\_EDMA3\_QUERY\_INTRPEND*,&regionIntr);

 } **while** (!(regionIntr.intr & 0x1));

 /\* Clear the pending bit \*/

 **CSL\_edma3HwControl**(hModule,*CSL\_EDMA3\_CMD\_INTRPEND\_CLEAR*,&regionIntr);

 /\* Close channel \*/

 **CSL\_edma3ChannelClose**(hChannel) ;

 /\* Close EDMA module \*/

 **CSL\_edma3Close**(hModule) != CSL\_SOK;

 /\* The test passed. \*/

}

**int** **main**(){

 **float** \*a = **malloc**(**sizeof**(**float**)\*4\*1024);//array a，put in DDR3

 **float** b[4096];//array b put in SHRAM(share ram)

 **int** i=0;

 //a initialization

 **for**(i=0;i<4096;i++)

 {

 a[i]=(**float**)i;

 **printf**("%f ",a[i]);

 **if**(i%1024==0 && i!= 0)

 **printf**("\n");

 }

 **printf**("\n");

 **printf**("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 //EDMA3 PARM set

 **int** aCnt,bCnt,cCnt,srcBidx,dstBidx,srcCidx,dstCidx;

 aCnt=4;//ACNT

 bCnt=1024;//bcnt

 cCnt=4;//CCNT

 srcBidx=aCnt;//srcBidx

 dstBidx=aCnt\*cCnt;//dstBidx

 srcCidx=aCnt\*bCnt;//srcCidx

 dstCidx=aCnt;//dstCidx

 //EMDA3

 Data\_Sorting(0,0,a,b,aCnt,

 bCnt,cCnt,srcBidx,dstBidx,srcCidx,dstCidx);

 // print result

 **for**(i=0;i<4096;i++)

 {

 **printf**("%f ",b[i]);

 **if**(i%4==0 && i!=0)

 **printf**("\n");

 }

 **return** 0;

}

The output of the simulator is as follows:



Question 1: There are some issues with the simulation data, and the transposed results are not quite correct. Is it a parameter issue?

Question 2: I am running this code on the C6678 development version and cannot output results. EDMA3 polling has fallen into a dead loop. What is the reason?

Looking forward to your reply! Thank you very much.