HLS LabB Cholesky algorithm

111061627 陳佳詳

在線性代數中，cholesky decomposition 是指將一個正定的 Hermiton 矩陣分解成一個下三角矩陣，與其共軛轉置之乘積，下圖為他的公式。

If we write out the equation

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} L_{11} & L_{21} & L_{31} \\ 0 & L_{22} & L_{32} \\ 0 & 0 & L_{33} \end{pmatrix} = \begin{pmatrix} L_{11}^2 & & \text{(symmetric)} \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{pmatrix}$$

we obtain the following:

$$\mathbf{L} = \begin{pmatrix} \sqrt{A_{11}} & 0 & 0 \\ A_{21}/L_{11} & \sqrt{A_{22} - L_{21}^2} & 0 \\ A_{31}/L_{11} & (A_{32} - L_{31}L_{21})/L_{22} & \sqrt{A_{33} - L_{31}^2 - L_{32}^2} \end{pmatrix}$$

and therefore the following formulas for the entries of **L**:

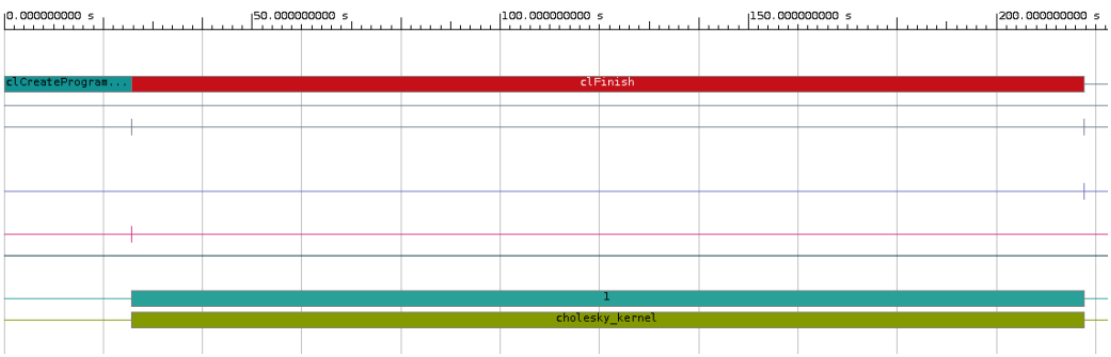
$$L_{j,j} = (\pm)\sqrt{A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2},$$

$$L_{i,j} = \frac{1}{L_{j,j}} \left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k} \right) \quad \text{for } i > j.$$

1. run this design in CPU

```
INFO: Matrix Row M: 512
INFO: Matrix Col N: 512
INFO: Finish CPU execution
INFO: CPU execution time is:15546 us
errA = 0
dataAN = 512
dataAM = 512
--------------
INFO: Result correct
```

2. module1 Baseline



Timing:

* Summary:

Clock	Target	Estimated	Uncertainty
ap_clk	3.33 ns	2.433 ns	0.90 ns

Latency:

* Summary:

Latency (cycles)		Latency (absolute)		Interval		Pipeline
min	max	min	max	min	max	Type
?	?	?	?	?	?	no

Utilization estimation:

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	294	-
FIFO	-	-	-	-	-
Instance	4	20	3561	3128	0
Memory	32	-	0	0	0
Multiplexer	-	-	-	2144	-
Register	-	-	1071	-	-
Total	36	20	4632	5566	0
Available SLR	1344	2976	871680	435840	320
Utilization SLR (%)	2	~0	~0	1	0
Available	2688	5952	1743360	871680	640
Utilization (%)	1	~0	~0	~0	0

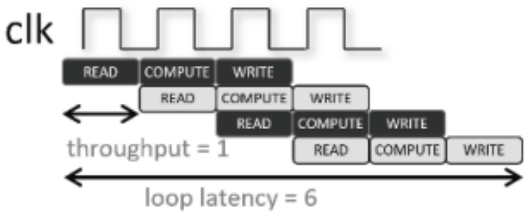
Interface:

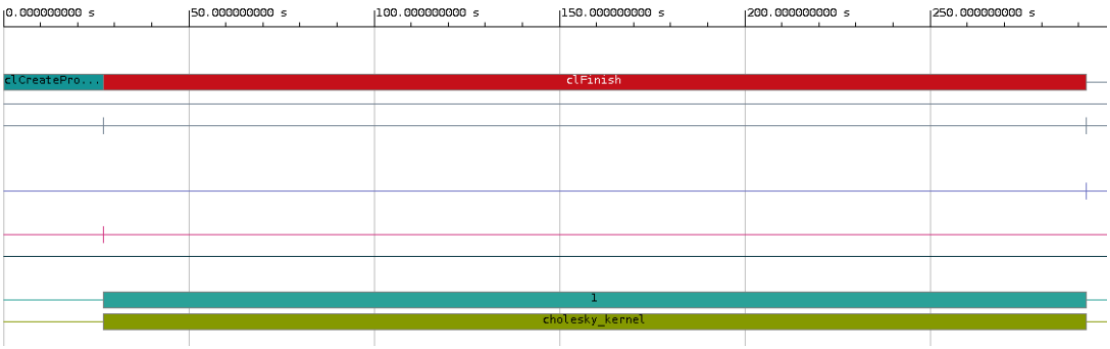
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	scalar
s_axi_control_AWREADY	out	1	s_axi	control	scalar
s_axi_control_AWADDR	in	6	s_axi	control	scalar
s_axi_control_WVALID	in	1	s_axi	control	scalar
s_axi_control_WREADY	out	1	s_axi	control	scalar
s_axi_control_WDATA	in	32	s_axi	control	scalar
s_axi_control_WSTRB	in	4	s_axi	control	scalar
s_axi_control_ARVALID	in	1	s_axi	control	scalar
s_axi_control_ARREADY	out	1	s_axi	control	scalar
s_axi_control_ARADDR	in	6	s_axi	control	scalar
s_axi_control_RVALID	out	1	s_axi	control	scalar
s_axi_control_RREADY	in	1	s_axi	control	scalar
s_axi_control_RDATA	out	32	s_axi	control	scalar
s_axi_control_RRESP	out	2	s_axi	control	scalar
s_axi_control_BVALID	out	1	s_axi	control	scalar
s_axi_control_BREADY	in	1	s_axi	control	scalar
s_axi_control_BRESP	out	2	s_axi	control	scalar
ap_local_block	out	1	ap_ctrl_chain	cholesky_kernel	return value
ap_clk	in	1	ap_ctrl_chain	cholesky_kernel	return value
ap_rst_n	in	1	ap_ctrl_chain	cholesky_kernel	return value
interrupt	out	1	ap_ctrl_chain	cholesky_kernel	return value
m_axi_gmem_AWVALID	out	1	m_axi	gmem	pointer
m_axi_gmem_AWREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_AWADDR	out	64	m_axi	gmem	pointer
m_axi_gmem_AWID	out	1	m_axi	gmem	pointer
m_axi_gmem_AWLEN	out	8	m_axi	gmem	pointer
m_axi_gmem_AWSIZE	out	3	m_axi	gmem	pointer
m_axi_gmem_AWBURST	out	2	m_axi	gmem	pointer
m_axi_gmem_AWLOCK	out	2	m_axi	gmem	pointer
m_axi_gmem_AWCACHE	out	4	m_axi	gmem	pointer
m_axi_gmem_AWPROT	out	3	m_axi	gmem	pointer
m_axi_gmem_AWQOS	out	4	m_axi	gmem	pointer
m_axi_gmem_AWREGION	out	4	m_axi	gmem	pointer
m_axi_gmem_AWUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_WVALID	out	1	m_axi	gmem	pointer
m_axi_gmem_WREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_WDATA	out	64	m_axi	gmem	pointer
m_axi_gmem_WSTRB	out	8	m_axi	gmem	pointer
m_axi_gmem_WLAST	out	1	m_axi	gmem	pointer
m_axi_gmem_WID	out	1	m_axi	gmem	pointer
m_axi_gmem_WUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_ARVALID	out	1	m_axi	gmem	pointer
m_axi_gmem_ARREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_ARADDR	out	64	m_axi	gmem	pointer
m_axi_gmem_ARID	out	1	m_axi	gmem	pointer
m_axi_gmem_ARLEN	out	8	m_axi	gmem	pointer
m_axi_gmem_ARSIZE	out	3	m_axi	gmem	pointer
m_axi_gmem_ARBURST	out	2	m_axi	gmem	pointer
m_axi_gmem_ARLOCK	out	2	m_axi	gmem	pointer
m_axi_gmem_ARCACHE	out	4	m_axi	gmem	pointer
m_axi_gmem_ARPROT	out	3	m_axi	gmem	pointer
m_axi_gmem_ARQOS	out	4	m_axi	gmem	pointer
m_axi_gmem_ARREGION	out	4	m_axi	gmem	pointer
m_axi_gmem_ARUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_RVALID	in	1	m_axi	gmem	pointer
m_axi_gmem_RREADY	out	1	m_axi	gmem	pointer
m_axi_gmem_RDATA	in	64	m_axi	gmem	pointer
m_axi_gmem_RLAST	in	1	m_axi	gmem	pointer
m_axi_gmem_RID	in	1	m_axi	gmem	pointer
m_axi_gmem_RUSER	in	1	m_axi	gmem	pointer
m_axi_gmem_RRESP	in	2	m_axi	gmem	pointer
m_axi_gmem_BVALID	in	1	m_axi	gmem	pointer
m_axi_gmem_BREADY	out	1	m_axi	gmem	pointer
m_axi_gmem_BRESP	in	2	m_axi	gmem	pointer
m_axi_gmem_BID	in	1	m_axi	gmem	pointer
m_axi_gmem_BUSER	in	1	m_axi	gmem	pointer

3. module2 pipeline

This module is meant to focus on the pipeline pragma and go through the description below. The kernel source code with the loops annotated with the pragma will produce the same results as in module 1, that's because since simple loops and inner loops (for nested loops) are automatically pipelined by the tool.

```
void F (...)  {
...
lbl: for (i=0;i=<4;i++) {
# PRAGMA HLS PIPELINE
      op_READ;
      op_COMPUTE;
      op_WRITE;
}
...
```





Performance estimation:

```
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  |  3.33 ns|  2.433 ns|     0.90 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |        ?|        ?|         ?|         ?|    ?|    ?|       no|
    +---------+---------+----------+----------+-----+-----+---------+
```

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	294	-
FIFO	-	-	-	-	-
Instance	4	20	3561	3128	0
Memory	32	-	0	0	0
Multiplexer	-	-	-	2144	-
Register	-	-	1135	-	-
Total	36	20	4696	5566	0
Available SLR	1344	2976	871680	435840	320
Utilization SLR (%)	2	~0	~0	1	0
Available	2688	5952	1743360	871680	640
Utilization (%)	1	~0	~0	~0	0

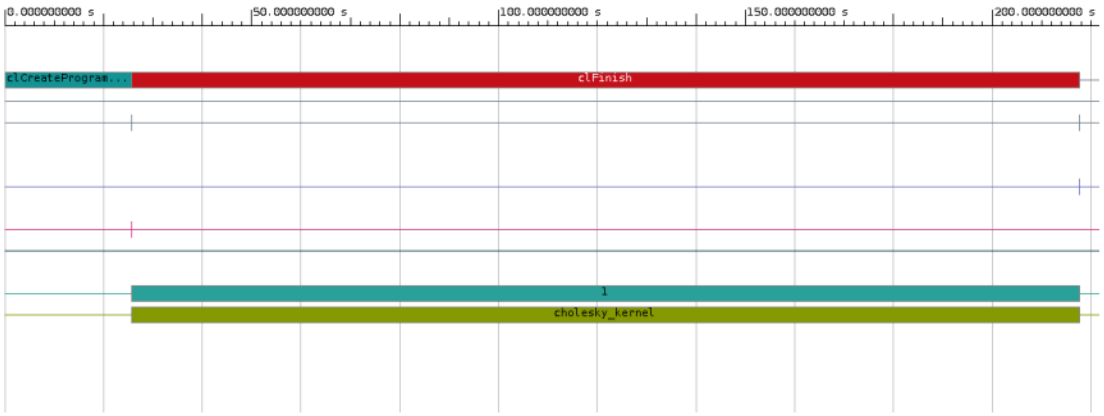
Interface:

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	scalar
s_axi_control_AWREADY	out	1	s_axi	control	scalar
s_axi_control_AWADDR	in	6	s_axi	control	scalar
s_axi_control_WVALID	in	1	s_axi	control	scalar
s_axi_control_WREADY	out	1	s_axi	control	scalar
s_axi_control_WDATA	in	32	s_axi	control	scalar
s_axi_control_WSTRB	in	4	s_axi	control	scalar
s_axi_control_ARVALID	in	1	s_axi	control	scalar
s_axi_control_ARREADY	out	1	s_axi	control	scalar
s_axi_control_ARADDR	in	6	s_axi	control	scalar
s_axi_control_RVALID	out	1	s_axi	control	scalar
s_axi_control_RREADY	in	1	s_axi	control	scalar
s_axi_control_RDATA	out	32	s_axi	control	scalar
s_axi_control_RRESP	out	2	s_axi	control	scalar
s_axi_control_BVALID	out	1	s_axi	control	scalar
s_axi_control_BREADY	in	1	s_axi	control	scalar
s_axi_control_BRESP	out	2	s_axi	control	scalar
ap_local_block	out	1	ap_ctrl_chain	cholesky_kernel	return value
ap_clk	in	1	ap_ctrl_chain	cholesky_kernel	return value
ap_rst_n	in	1	ap_ctrl_chain	cholesky_kernel	return value
interrupt	out	1	ap_ctrl_chain	cholesky_kernel	return value

m_axi_gmem0_AWVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_AWADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_AWID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_AWSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_WDATA	out	64	m_axi	gmem0	pointer
m_axi_gmem0_WSTRB	out	8	m_axi	gmem0	pointer
m_axi_gmem0_WLAST	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_ARADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_ARID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_ARSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RDATA	in	64	m_axi	gmem0	pointer
m_axi_gmem0_RLAST	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RUSER	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_BRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BUSER	in	1	m_axi	gmem0	pointer

4. module 3 datatype

In this module both the kernel and host code are modified to use 32-bit floating point data types(float) instead of the 64-bit floating point(double) to show the performance and Xilinx utilization beneficial impact of downsizing data types.



== Performance Estimates

+ Timing:
    * Summary:

Clock	Target	Estimated	Uncertainty
ap_clk	3.33 ns	2.433 ns	0.90 ns

+ Latency:
    * Summary:

Latency (cycles)		Latency (absolute)		Interval		Pipeline
min	max	min	max	min	max	Type
?	?	?	?	?	?	no

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	294	-
FIFO	-	-	-	-	-
Instance	2	14	2496	2388	0
Memory	16	-	0	0	0
Multiplexer	-	-	-	1820	-
Register	-	-	930	-	-
Total	18	14	3426	4502	0
Available SLR	1344	2976	871680	435840	320
Utilization SLR (%)	1	~0	~0	1	0
Available	2688	5952	1743360	871680	640
Utilization (%)	~0	~0	~0	~0	0

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	scalar
s_axi_control_AWREADY	out	1	s_axi	control	scalar
s_axi_control_AWADDR	in	6	s_axi	control	scalar
s_axi_control_WVALID	in	1	s_axi	control	scalar
s_axi_control_WREADY	out	1	s_axi	control	scalar
s_axi_control_WDATA	in	32	s_axi	control	scalar
s_axi_control_WSTRB	in	4	s_axi	control	scalar
s_axi_control_ARVALID	in	1	s_axi	control	scalar
s_axi_control_ARREADY	out	1	s_axi	control	scalar
s_axi_control_ARADDR	in	6	s_axi	control	scalar
s_axi_control_RVALID	out	1	s_axi	control	scalar
s_axi_control_RREADY	in	1	s_axi	control	scalar
s_axi_control_RDATA	out	32	s_axi	control	scalar
s_axi_control_RRESP	out	2	s_axi	control	scalar
s_axi_control_BVALID	out	1	s_axi	control	scalar
s_axi_control_BREADY	in	1	s_axi	control	scalar
s_axi_control_BRESP	out	2	s_axi	control	scalar
ap_local_block	out	1	ap_ctrl_chain	cholesky_kernel	return value
ap_clk	in	1	ap_ctrl_chain	cholesky_kernel	return value
ap_rst_n	in	1	ap_ctrl_chain	cholesky_kernel	return value
interrupt	out	1	ap_ctrl_chain	cholesky_kernel	return value
m_axi_gmem0_AWVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_AWADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_AWID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_AWSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_WDATA	out	32	m_axi	gmem0	pointer
m_axi_gmem0_WSTRB	out	4	m_axi	gmem0	pointer
m_axi_gmem0_WLAST	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_ARADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_ARID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_ARSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RDATA	in	32	m_axi	gmem0	pointer
m_axi_gmem0_RLAST	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RUSER	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_BRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BUSER	in	1	m_axi	gmem0	pointer

5. module4 dataflow

The DATAFLOW pragma enables task-level pipelining, allowing functions and loops to overlap in their operation, increasing the concurrency of the register transfer level(RTL) implementation, and increasing the overall throughput of the design.

```
wr_loop_j: for (int j = 0; j < TILE_PER_ROW; ++j) {
#pragma HLS DATAFLOW
   wr_buf_loop_m: for (int m = 0; m < HEIGHT; ++m) {
      wr_buf_loop_n: for (int n = 0; n < WIDTH; ++n) {
      #pragma HLS PIPELINE
      // should burst WIDTH in WORD beat
         outFifo >> tile[m][n];
      }
   }
   wr_loop_m: for (int m = 0; m < HEIGHT; ++m) {
      wr_loop_n: for (int n = 0; n < WIDTH; ++n) {
      #pragma HLS PIPELINE
         outx[HEIGHT*TILE_PER_ROW*WIDTH*i+TILE_PER_ROW*WIDTH*m+WIDTH*j+n] = tile[m][n];
      }
   }
}
```

以下程式會呼叫 chol_col 16 次

```
template <typename T, int N, int NCU>
void chol_col_wrapper(int n, T dataA[NCU][(N + NCU - 1) / NCU][N], T dataj[NCU][N], T tmp1, int j)
{
#pragma HLS DATAFLOW

Loop_row:
    for (int num = 0; num < NCU; num++)
    {
#pragma HLS unroll factor = NCU
        chol_col<T, N, NCU>(n, dataA[num], dataj[num], tmp1, num, j);
    }
}
```

== Performance Estimates

+ Timing:
    * Summary:

Clock	Target	Estimated	Uncertainty
ap_clk	3.33 ns	2.433 ns	0.90 ns

+ Latency:
    * Summary:

Latency (cycles)		Latency (absolute)		Interval		Pipeline
min	max	min	max	min	max	Type
?	?	?	?	?	?	no

== Utilization Estimates

* Summary:

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	88	-
FIFO	-	-	-	-	-
Instance	4	196	81560	60228	0
Memory	32	-	0	0	64
Multiplexer	-	-	-	3189	-
Register	-	-	631	-	-
Total	36	196	82191	63505	64
Available SLR	1344	2976	871680	435840	320
Utilization SLR (%)	2	6	9	14	20
Available	2688	5952	1743360	871680	640
Utilization (%)	1	3	4	7	10

== Interface

* Summary:

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	scalar
s_axi_control_AWREADY	out	1	s_axi	control	scalar
s_axi_control_AWADDR	in	6	s_axi	control	scalar
s_axi_control_WVALID	in	1	s_axi	control	scalar
s_axi_control_WREADY	out	1	s_axi	control	scalar
s_axi_control_WDATA	in	32	s_axi	control	scalar
s_axi_control_WSTRB	in	4	s_axi	control	scalar
s_axi_control_ARVALID	in	1	s_axi	control	scalar
s_axi_control_ARREADY	out	1	s_axi	control	scalar
s_axi_control_ARADDR	in	6	s_axi	control	scalar
s_axi_control_RVALID	out	1	s_axi	control	scalar
s_axi_control_RREADY	in	1	s_axi	control	scalar
s_axi_control_RDATA	out	32	s_axi	control	scalar
s_axi_control_RRESP	out	2	s_axi	control	scalar
s_axi_control_BVALID	out	1	s_axi	control	scalar
s_axi_control_BREADY	in	1	s_axi	control	scalar
s_axi_control_BRESP	out	2	s_axi	control	scalar
ap_local_block	out	1	ap_ctrl_chain	cholesky_kernel	return value
ap_clk	in	1	ap_ctrl_chain	cholesky_kernel	return value
ap_rst_n	in	1	ap_ctrl_chain	cholesky_kernel	return value
interrupt	out	1	ap_ctrl_chain	cholesky_kernel	return value
m_axi_gmem0_AWVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_AWADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_AWID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_AWLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_AWSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_AWCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_AWQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_AWUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_WDATA	out	64	m_axi	gmem0	pointer
m_axi_gmem0_WSTRB	out	8	m_axi	gmem0	pointer
m_axi_gmem0_WLAST	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_WUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARVALID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARREADY	in	1	m_axi	gmem0	pointer
m_axi_gmem0_ARADDR	out	64	m_axi	gmem0	pointer
m_axi_gmem0_ARID	out	1	m_axi	gmem0	pointer
m_axi_gmem0_ARLEN	out	8	m_axi	gmem0	pointer
m_axi_gmem0_ARSIZE	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARBURST	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARLOCK	out	2	m_axi	gmem0	pointer
m_axi_gmem0_ARCACHE	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARPROT	out	3	m_axi	gmem0	pointer
m_axi_gmem0_ARQOS	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARREGION	out	4	m_axi	gmem0	pointer
m_axi_gmem0_ARUSER	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_RDATA	in	64	m_axi	gmem0	pointer
m_axi_gmem0_RLAST	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RUSER	in	1	m_axi	gmem0	pointer
m_axi_gmem0_RRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BVALID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BREADY	out	1	m_axi	gmem0	pointer
m_axi_gmem0_BRESP	in	2	m_axi	gmem0	pointer
m_axi_gmem0_BID	in	1	m_axi	gmem0	pointer
m_axi_gmem0_BUSER	in	1	m_axi	gmem0	pointer

Result Summary:

Module	CPU	Module1	Moduel2	Module3	Module4
Exe. Time	21461	793950	793732	536784	11698
Speed up(cpu)	1	0.03x	0.03x	0.04x	1.83x
Speed up	N/A	1	1	1.48x	68x