22 2Syst (Matrix Multiplication[1])

The filter circuit of this section and the filter of section 21 have been chosen as witnesses of one-dimensional and two-dimensional systolic architectures. The notion of "systolic arrays" has first been introduced by Kung and Leierson [KuLe78]. Basics on these specialized regular architectures can be found in [Kung82].

Systolic arrays require additional efforts for specifying the indented behavior, since here data have to be applied at different times in a special order to achieve a correct functioning of the circuit. Moreover, more-dimensional systolic arrays are a good illustration of more-dimensional generic circuits.

22.1 Specification

22.1.1 General Specification

The circuit to be presented here is a two-dimensional "hexagonal" systolic array. The circuit has been taken from [MeCo80] and is intended to multiply two matrices.

The multiplication of two $n \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$ results in a matrix $C = (c_{ij})$, where the multiplication is defined by the recursive equation 22-1.

$$\begin{aligned} c_{ij}^{(1)} &= 0 \\ c_{ij}^{(k+1)} &= c_{ij}^{(k)} + a_{ik} b_{kj} \\ c_{ij} &= c_{ij}^{(n+1)} \end{aligned} \tag{22-1}$$

However, the matrix multiplier must only be able to multiply so-called "band" matrices. These are special matrices where certain matrix positions must carry zeros. The multiplication of two 3×3 and two 4×4 matrices *A* and *B*, having width $w = 3$ and $w = 4$ are given in equation 22-2 and equation 22-3, respectively (note that A and B have the zeros at "mirrored" places).

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & 0 \\ b_{21} & b_{22} & b_{23} \\ 0 & b_{22} & b_{33} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{22-2}$$

$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & b_{13} & 0 \\ b_{21} & b_{22} & b_{23} & b_{24} \\ 0 & b_{32} & b_{33} & b_{34} \\ 0 & 0 & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \tag{22-3}$$

1. The VDHL files often refer to this circuit as MATRIX.

22.1.2 Timing Requirements

The black box diagram of the circuit for a 4×4 multiplication is shown in Fig. 22-1.

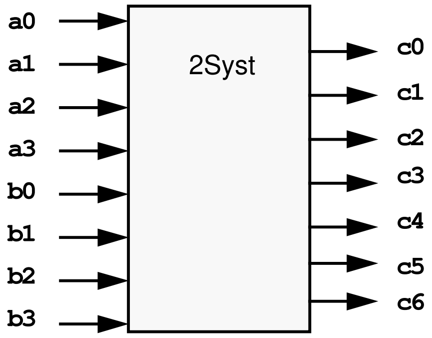

Figure 22-1: Black bock view of 2Syst

For performing a 4×4 multiplication with a band width $w = 4$, the values a_{ij} and b_{ij} have to be fed into the inputs **a0** to **b3** at the time instants as indicated in table 22-1.

t	a3	a2	a1	a0	b3	b2	b1	b0	c6	c5	c4	c3	c2	c1	c0
0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-
1	0	0	a_{11}	0	0	0	b_{11}	0	-	-	-	-	-	-	-
2	0	a_{21}	0	0	0	b_{12}	0	0	-	-	-	-	-	-	-
3	a_{31}	0	0	a_{12}	b_{13}	0	0	b_{21}	0	0	0	c_{11}	0	0	0
4	0	0	a_{22}	0	0	0	b_{22}	0	0	0	c_{21}	0	c_{12}	0	0
5	0	a_{32}	0	0	0	b_{23}	0	0	0	c_{31}	0	0	0	c_{13}	0
6	a_{42}	0	0	a_{23}	b_{24}	0	0	b_{32}	c_{41}	0	0	c_{22}	0	0	c_{14}
7	0	0	a_{33}	0	0	0	b_{33}	0	0	0	c_{32}	0	c_{23}	0	0
8	0	a_{43}	0	0	0	b_{34}	0	0	0	c_{42}	0	0	0	c_{24}	0
9	0	0	0	a_{34}	0	0	0	b_{43}	0	0	0	c_{33}	0	0	0
10	0	0	a_{44}	0	0	0	b_{44}	0	0	0	c_{43}	0	c_{34}	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	c_{44}	0	0	0
13	0	0	a_{11}	0	0	0	b_{11}	0	0	0	0	0	0	0	0
14	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Table 22-1: Input/output timing scheme

Obviously one of the input matrices is fed column by comun, the other row by row into the circuit. The diagonal elements a_{ii} and b_{jj} are fed into the $\left\lfloor \frac{w-1}{2} \right\rfloor$-th input (**a1** and **b1** in case of $w = 4$, **a2** and **b2** in case of $w = 5$). The time distance between input columns and rows is 2. Output values are available after 2 time instances. The mapping of output matix values c_{ij} to output signals ck ist given in Fig. 22-2. The distance between consecutive output values is 2.

The total length of the output stream equals the length of the input stream, hence it is finished 2 time instances after the last input value has been applied.

The outputs c0 to c6 produce the result c_{ij} according to the scheme given in table 22-2.

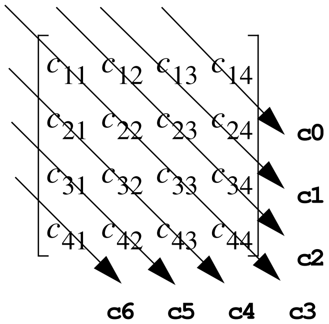

Figure 22-2: Output variable correspondence

22.2 Implementation

22.2.1 General architecture

The overall hexagonal architecture of the systolic array for an 4×4 example is given in Fig. 22-3.

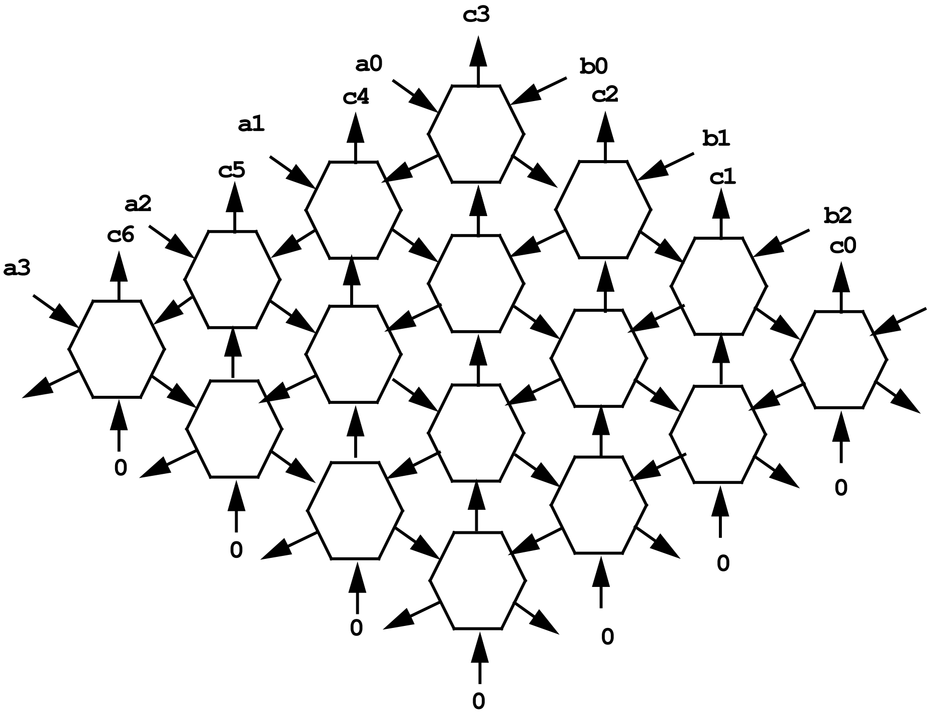

Figure 22-3: Basic array structure

22.2.2 Implementation of one cell

Each cell of Fig. 22-3 has three inputs `A_In<n-1:0>`, `B_In<n-1:0>` and `C_In<m-1:0>` and three outputs `A_Out<n-1:0>`, `B_Out<n-1:0>` and `C_Out<m-1:0>` (Fig. 22-4).

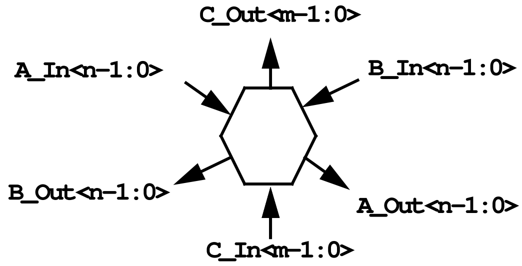

Figure 22-4: Black box view of one cell

The inputs `A_In<n-1:0>` and B`_In<n-1:0>`are connected to a *n* bit register and the input of a multiplier. The 2*n* bit multiplier output is connected to an adder. The second input of the adder is fed by `C_In<m-1:0>`. The adder output is fed in a register. The register outputs provide the cell outputs `A_Out<n-1:0>`, `B_Out<n-1:0>` and `C_Out<m-1:0>` (Fig. 22-5).

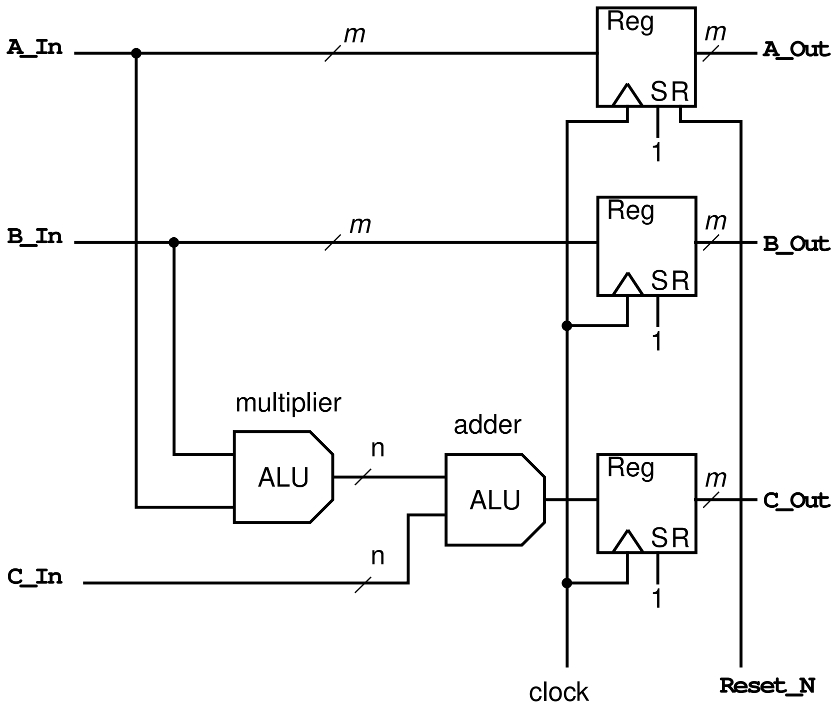

Figure 22-5: Realization of one array cell

22.2.3 Actual Realization and Example

A 4×4 realization of the basic cell (Fig. 22-5) and the whole systolic array (Fig. 22-3) is given in Fig. 22-6 and Fig. 22-7, respectively.

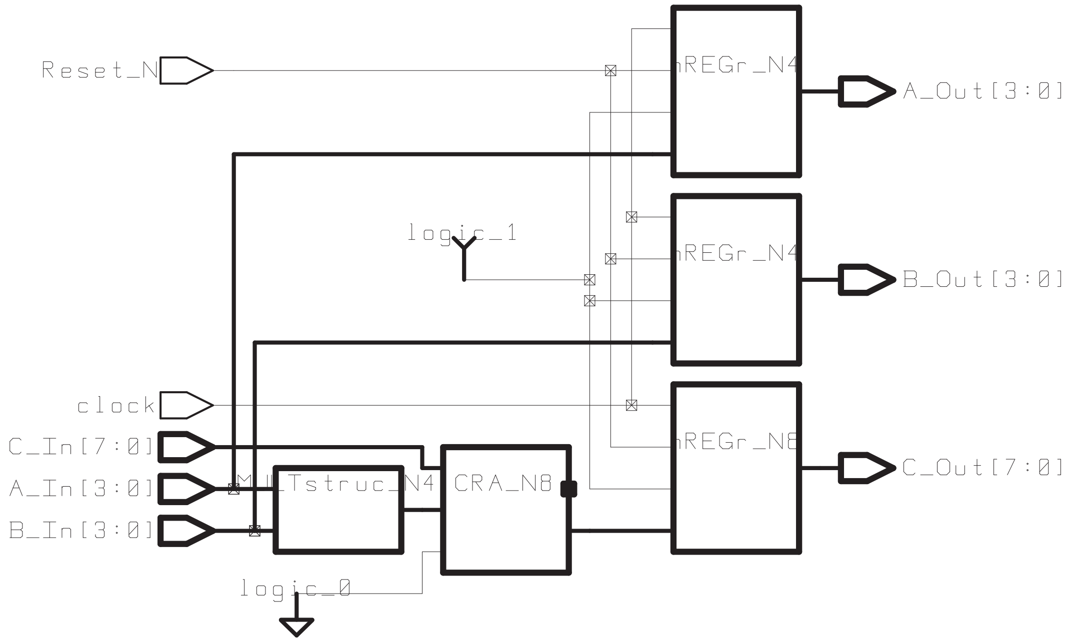

Figure 22-6: SYNOPSYS realization of a basic cell

The following waveform (Fig. 22-8) shows the result of computing the multiplication given in equation 22-4.

$$\begin{bmatrix} 3 & 2 & 0 & 0 \\ 5 & 1 & 3 & 0 \\ 7 & 4 & 2 & 5 \\ 0 & 2 & 1 & 8 \end{bmatrix} \cdot \begin{bmatrix} 4 & 2 & 1 & 0 \\ 5 & 3 & 1 & 2 \\ 0 & 6 & 2 & 7 \\ 0 & 0 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 22 & 12 & 5 & 4 \\ 25 & 31 & 12 & 23 \\ 48 & 38 & 35 & 47 \\ 10 & 12 & 36 & 51 \end{bmatrix} \qquad (22\text{-}4)$$

The busses **`StrA0In`** to **`StrA3In`** and **`StrB0In`** to **`StrB3In`** carry the coefficients of the matrices A and B.

22.3 Status and Acknowledgments

Thanks to H.-P. Eich and C.-J. Thomas for designing the circuit in a commercial design system.

The documentation of 2Syst has been completely revised September 1995 due to valuable comments from Scott Hazelhurst, University of Brithish Columbia, Canada.

22.4 Literature

[MeCo80] C. Mead and L. Conway. *Introduction to VLSI Design*. Addison-Wesley, 2. edition, October 1980.

[Fram92] CADENCE Design Framework II version 4.2a. Reference Manual, February 1992.

[Kung82] H.T. Kung. Why systolic architectures. *IEEE Computer*, pages 37–46, January 1982.

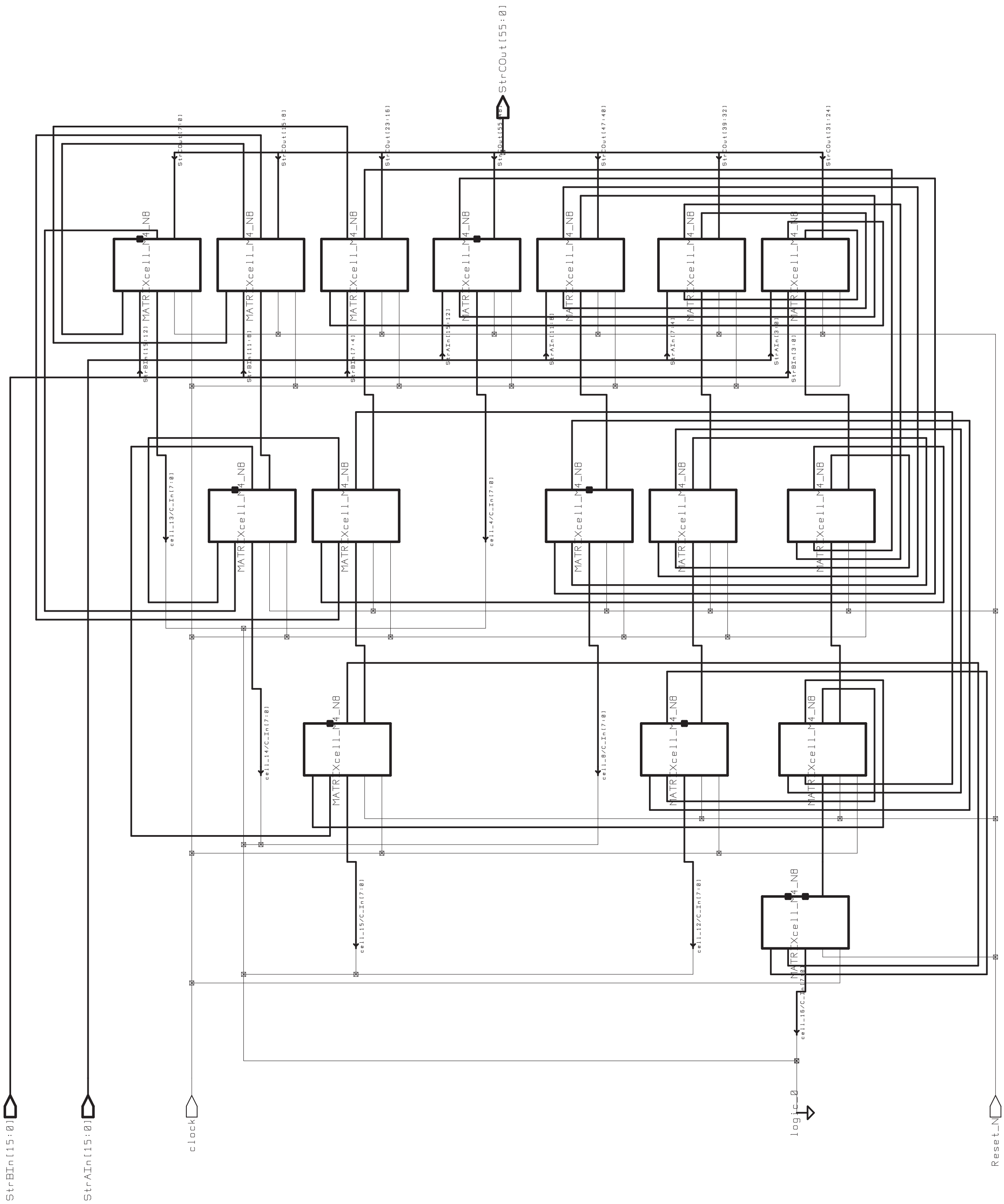

Figure 22-7: Realization of a 4×4 array

[KuLe78] H.T. Kung and C.E. Leierson. Systolic arrays (for VLSI). In *Sparse Matrics Proceedings*, pages 256–282. Society for Industrial and Applied Mathematics 1979, 1978.

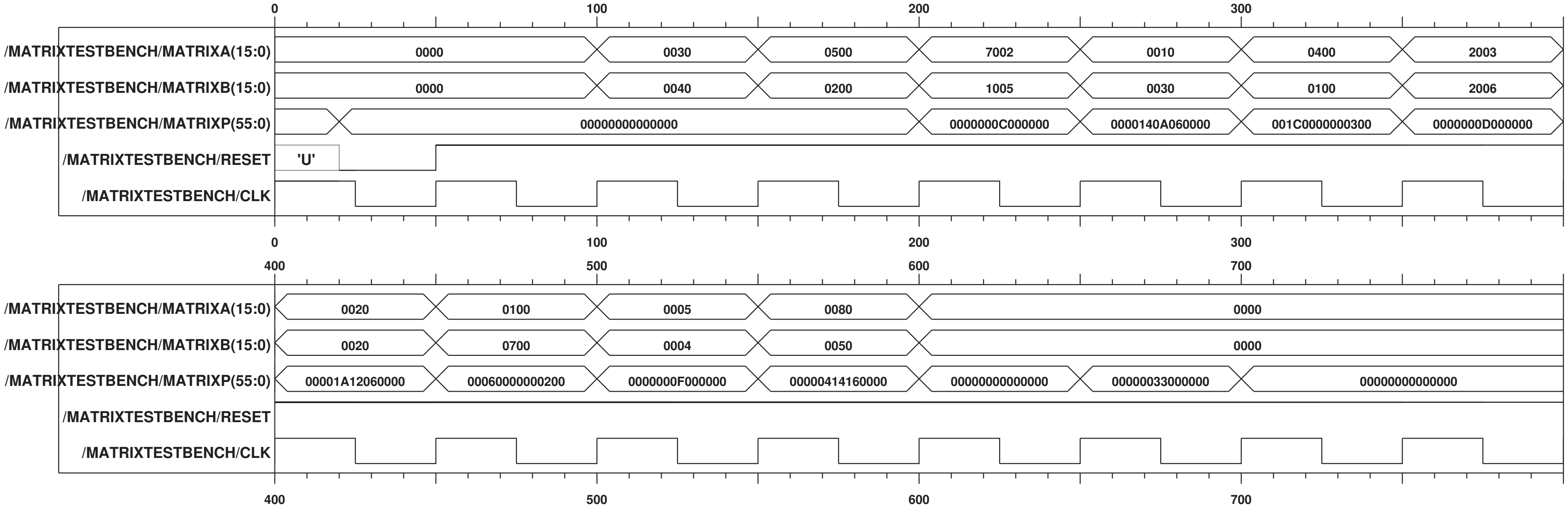

Figure 22-8: Waveform of computing equation 22-4