International Journal of Electronics Communication and Computer Engineering
Volume 10, Issue 3, ISSN (Online): 2249–071X

Data Access Time Reduction for 3D Data Bus in Mixed Cases of Embedded Bus Switches and Inserted Signal Repeaters

Chia-Chun Tsai
Dept. of Computer Science and Information Engineering Nanhua University, Chiayi, Taiwan (R.O.C.)
Corresponding author email id: chun@nhu.edu.tw

Date of publication (dd/mm/yyyy): 24/06/2019

Abstract **– Data are frequently running on the 3D data bus of a stacked-layer chip at different timing periods, their data access time dominates the chip performance. In this paper, we proposed an integrated algorithm that combines a few methodologies for data access time reduction. The first strategy is to embed bus switches into data bus to isolate those unnecessary across local bus capacitive loadings accessed at different timing periods. Their critical access time and average access time can be obviously reduced. Moreover, two other strategies, sizing all the source drivers and inserting repeaters into bus wires, can reduce their access time in advance to a given 3D data bus with embedded bus switches. The integrate algorithm can work with an independent strategy or any mixed cases of combining different strategies for minimize their access time. Experimental results show their comparison. The critical access time and average access time of a 3D data bus with embedded bus switches are reduced up to 54.56% and 60.53%, respectively, on average. For the case of inserting repeaters into bus wires to a given data bus with embedded bus switches, the reduced rate of critical access time and average access time are 77.99% and 68.15%, respectively, in running time of 0.747 seconds and repeater sizes of 107 on average.**

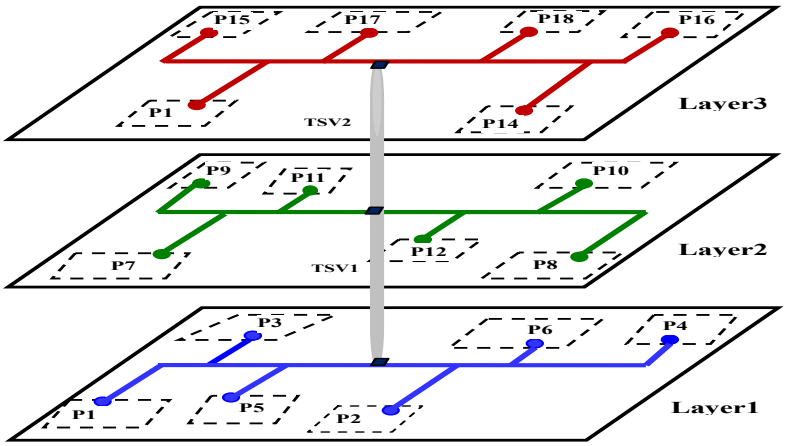
Keywords **– 3D Data Bus, Data Access Time, TSV, Bus Switch, Signal Repeater.**

I. INTRODUCTION

A 3D stacked-layer chip is very popular for constructing a complex SoC (System-on a Chip). The stacked-layer chip is a multiple-layer integrated circuit that can be made with various CMOS nanometer technologies [1]. Each layer has at least one die which consists of a number of different function-based modules and has a local bus combining the control bus, address bus, and data bus to interact and/or connect all the modules. A number of TSVs (Through Silicon Vias) [2] is used to vertically connect and/or contact all the buses located on each layer. These 2D local buses are integrated to be a global system bus. Here, we focus the data bus and concern the performance of data access on the bus. All the 2D local data buses are tightly formed to be a 3D data bus. Data are frequently running on the 3D data bus and their data access time will dominate the performance of a 3D stacked-layer chip. Therefore, improving the access time of data ran on a 3D data bus is very important.

Fig. 1 shows a typical topology of 3D data bus in a 3D stacked-layer chip. The 3D data bus has eighteen terminals (i.e., p1 to p18), thirty-eight bus wires, and fifty-five data access timings at eighteen timing periods (i.e., TP-1 to TP-18) ran on the bus. Each layer consists of six functional modules and has its own local data bus. Three local data buses are connected and interacted via two TSVs. Data can be transferred from one module to other modules via the 3D data bus and/or TSVs at a particular timing period. For example, data can be propagated from source terminal p1 to four sink terminals p2, p4, p14, and p16 at the timing period TP-1. The timing period TP-1 has four access timings, $p1$-$p2$, $p1$-$p4$, $p1$-$p14$, and $p1$-$p16$, and four access times, $T_{p1\text{-}p2}$, $T_{p1\text{-}p4}$, $T_{p1\text{-}p14}$, and $T_{p1\text{-}p16}$. Another data can also be transferred from the source terminal p10 to two sink terminals p8 and p15 at the other timing period TP-10. The timing period TP-10 has two access times. Notably, there is only one source and at least

one sink for data access at each timing period. A terminal can be viewed as a source or sink at a non-overlapping timing period. For a terminal, e.g., p1, it is a driving source at timing period TP-1 but is also a loading sink at another timing period TP-12. Accordingly, a 3D data bus has the characteristics of multiple-source multiple-sink bidirectional transmission in nature.



Timing-Period	Source	Sinks	Timing- Period	Source	Sinks
TP-1	p1	p2, p4, p10, p16	TP-10	p10	p8, p15
TP-2	p2	p1, p3, p9	TP-11	p11	p6, p18
TP-3	p3	p1, p2, p4, p8, p13	TP-12	p12	p1, p5, p17
TP-4	p4	p3, p15	TP-13	p13	p3, p14, p15, p16
TP-5	p5	p11, p13	TP-14	p14	p10, p13, p16
TP-6	p6	p11, p17	TP-15	p15	p4, p8, p14, p16
TP-7	p7	p1, p8, p9	TP-16	p16	p7, p14, p15
TP-8	p8	p3, p7, p9, p10, p15	TP-17	p17	p4, p12
TP-9	p9	p2, p7, p10, p14	TP-18	p18	p5, p11

Fig. 1. A typical topology of 3D data bus including fifty-five data access timings at eighteen timing periods.

The *critical access time* of a 3D data bus is defined as the maximum access time propagated from the path of source *i* to sink *j* at a timing period. The *average access time* of a 3D data bus is also defined as that the sum of data access times is divided by the total access timings at all the timing periods. The performance of a stacked-layer chip depends on both critical access time and average access time of a data bus. As shown in Fig. 1, the 3D data bus has fifty-five access times associated with eighteen timing periods and has a critical access time and an average access time. These two access times of a 3D data bus dominate the chip performance.

For one of data access timings, data may be accessed across the same layer, adjacent one layer, or adjacent two layers. As shown in Fig. 1, there are 55 data access timings that the number of data access timings accessed across 0, 1 and 2 layers are 24, 21 and 10. For the data access timings accessed across 0 layer, data only run on their own local bus and the unnecessary other local buses will be caused extra capacitive loadings for the data access time. Fig. 2(a) shows the fact that the total capacitance of the 1st local data bus, C_{BUS1}, is an extra capacitive loading while data are transferred from source p10 located on the 2nd local bus to sink p15 located on the 3rd local bus at the timing period TP-10. Similarly, the total capacitance of the 2nd local bus, C_{BUS2}, is an extra capacitive loading in Fig. 2(b) while their access time is computed from source p1 located on the 1st local bus to sink p14 or p16 located on the 3rd local bus at another timing period TP-1. Meanwhile, if bus switches can be embedded to isolate these extra capacitive loadings, their data access time will be obviously reduced.

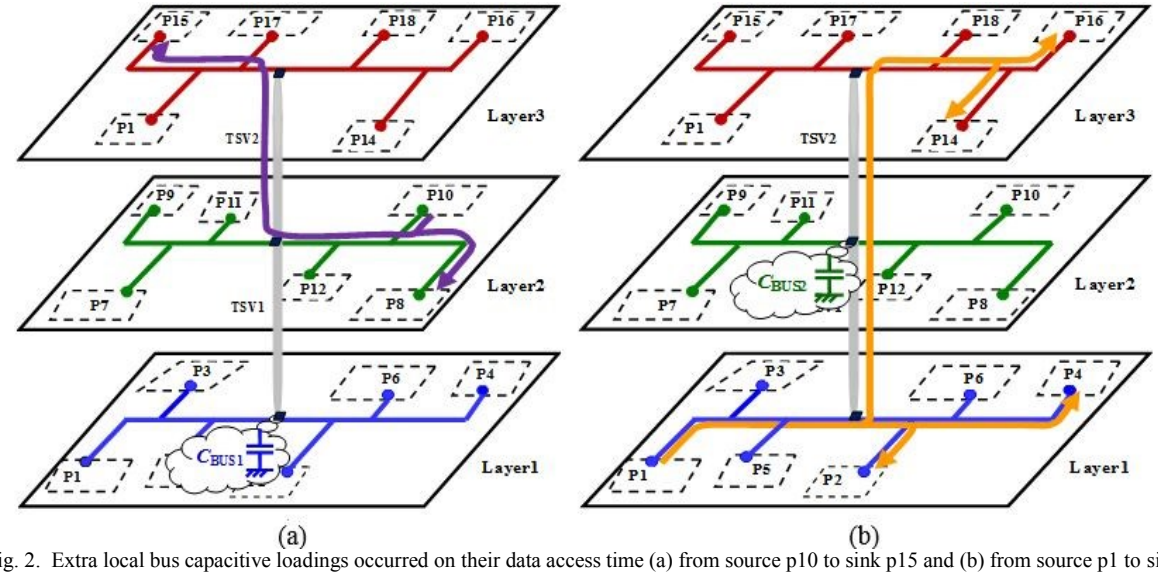


Fig. 2. Extra local bus capacitive loadings occurred on their data access time (a) from source p10 to sink p15 and (b) from source p1 to sink p14 or p16.

Many literatures were introduced about the different meaningful isolation of a data bus. Daneshtalab et al. [3] proposed a high-performance inter-layer bus structure (HIBS) for 3D stacked layers. The HIBS can reduce the complexity of bus arbitrators and save the propagation delay of data communication. Thakkar et al. [4] introduced a new architecture called 3D-Wiz for reducing the interaction overloading between data bus of DRAMs. The architecture can reduce the access time to any DRAMs. Cho et al. [2] presented the analysis of system bus on a multi-layer SoC considering the interconnection of TSVs. They found the maximum throughput of 3D system bus depending on the data bandwidth and number of layers. Tsai [5] first conducted repeater insertion to minimize the propagation delay for a 3D data bus based on RC delay model. The authors improved their strategies for critical access time minimization [6-7], but they do not to consider the capacitive loading effect for a local data bus which is not accessed. Tsai [8] created the concept of inserting isolated switches for data bus, but they just consider the effectiveness of source drivers for access time.

This work we propose an integrate algorithm for data access time reduction in mixed cases of embedded bus switches and inserted repeaters. We refer two published papers [8-9] and design various cases by combining the sized source drivers or inserted signal repeaters associated with embedded bus switches. These cases are: a 3D data bus without/ with embedded bus switches, sized source drivers for a given 3D data bus with embedded bus switches, inserted signal repeaters for a given 3D data bus with embedded bus switches, and sized source drivers and inserted repeaters for a given 3D data bus with embedded bus switches. Their demonstrated results show the comparison in the reduction of their critical access time and average access time for a 3D stacked-layer chip.

II. Problem Formulation

For a typical 3D stacked-layer chip, the chip performance depends on the transferred speed of data access on the 3D data bus. That is, their critical data access time and average data access time dominate the chip performance. Thus, the problem of data access time reduction to a 3D data bus can be formulated as follow.

Given the topology of a 3D data bus that has m terminals, n bus wires, p different timing periods, and each timing period has at least one data access timing transferred from a source to at least one sink, the objective is to reduce the critical access time as well as the average access time.

III. Integrated Algorithm for Data Access Time Reduction

We design various cases to divide the above problem into a number of sub-problems or problem cases and propose an integrated algorithm to solve their problem cases. These problem-case definitions and their solutions are presented as follows.

A. *Algorithm1 for Problem-Case1: Embedded Bus Switches into Data Bus*

A bus switch has capable to isolate the unused bus capacitive loading, but the high-speed performance of their switch-on or switch-off must be absolutely required. Many companies have very good bus switches, IDT [10] proposed 8-bit data bus quick-bidirectional-switch using CMOS process and Toshiba [11] investigated quad bus-buffer switch using C^2MOS process for local bus isolation. We adopt the latest bus switch referring 45nm predictive technology model [12]. Fig. 3(a) shows a bidirectional bus switch TG that the switch-on resistance r_G, input capacitance c_G, and propagation delay t_G are 1Ω, 1pF, and 10ps, respectively. Fig. 3(b) transfers data from A to B whereas Fig. 3(c) transfers from B to A. However, Fig. 3(d) isolates data transmission between two sides. Basically, the embedded bus switch may increase a little access time, but their side effect is very small due to its high-speed switched characteristics.

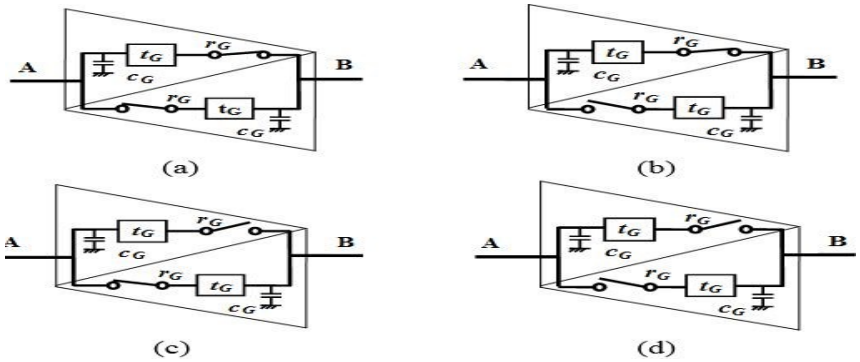


Fig. 3. The equivalent bus switches of (a) bidirectional transmission, (b) A to B, (c) B to A, and (d) two-side isolation.

Figs. 4(a) and 4(b) present two cases of embedding bus switches that can isolate extra local bus capacitive loadings from Figs. 2(a) and 2(b), respectively. A bus switch is inserted into the location between a TSV and a local bus. This approach is very helpful for isolating the unnecessary data bus capacitive loading and reducing their critical access time and average access time.

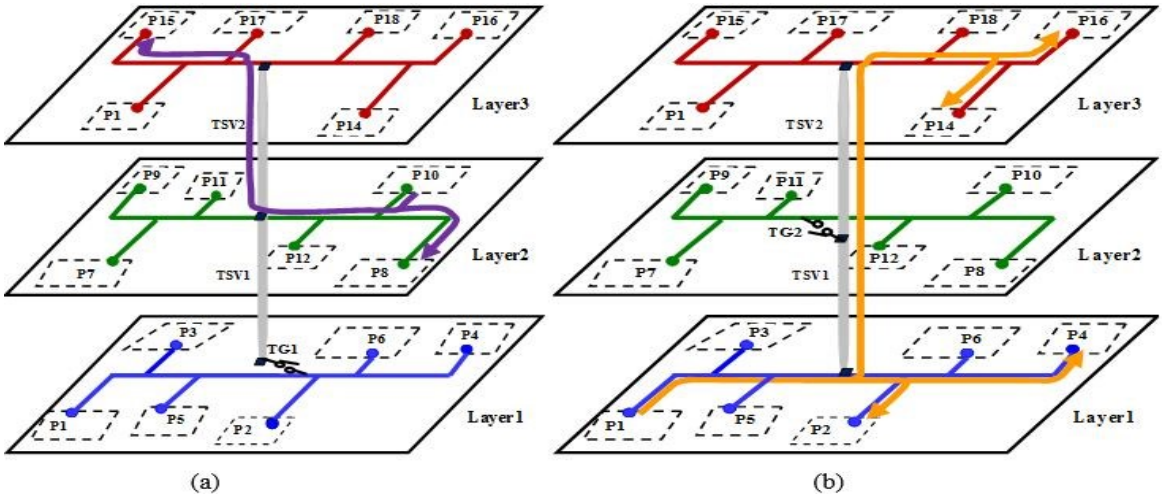


Fig. 4. Embedded bus switches isolate extra local bus capacitive loadings, (a) from source p10 to sink p15 and (b) from source p1 to sink p14 or p16, for reducing the critical access time.

The objective of the *Problem-Case1* is to embed bus switches for access time reduction. Fig. 5 shows the *Algorithm1* to solve the *Problem-Case1*. The function Find_AccessTime() is used to find the access time from a source to a sink. For a data bus without any embedded bus switches, the critical access time and average access time are *wo-Tcri* and *wo-Av_Tcri*, respectively. *w-Tcri* and *w-Av_Tcri* are the critical access time and average access time with embedded bus switches, respectively. Expectably, *w-Tcri* is less than *wo-Tcri* while *w-Av_Tcri* is also less than *wo-Av_Tcri*.

```
Algorithm1: DataBus_Embedded_BusSwitches
{   /* A bus topology with the number of m terminals, n segments, and p
       timing periods (p*h is the number of total timings). */
   Scan a 3D data bus topology and construct its data structure.
   wo-Tcri = 0; Sum = 0;
   for (each access timing)    // without embedded bus switches
   { AccessTime = Find_AccessTime(); Sum = Sum+AccessTime;
     if ( AccessTime > wo-Tcri)        wo_Tcri = AccessTime;
   }
   wo-Av_Tcri = Sum / (p*h);
   Embed bus switches between TSVs and local buses.
   w_Tcri = 0; Sum = 0;
   for (each access timing)    // with embedded bus switches
   { AccessTime = Find_AccessTime(); Sum = Sum+AccessTime;
     if ( AccessTime > w_Tcri)         w_Tcri = AccessTime;
   }
   w-Av_Tcri = Sum / (p*h);
   Return the critical access time w-Tcri and average access time w-Av_Tcri
}
```

Fig. 5. Algorithm1 is for access time reduction to a 3D data bus with embedded bus switches.

B. *Algorithm2 for Problem-Case2: Sized Source Drivers for a Given Data Bus with Embedded Bus Switches*

Based on the data bus with embedded bus switches, their critical access time and average access time can be reduced in advance only by sizing their source drivers. Figs. 6(a) and 6(b) show two cases of sizing the source drivers, p10 and p1, to their sinks, respectively, at different timing periods for access time reduction in advance.

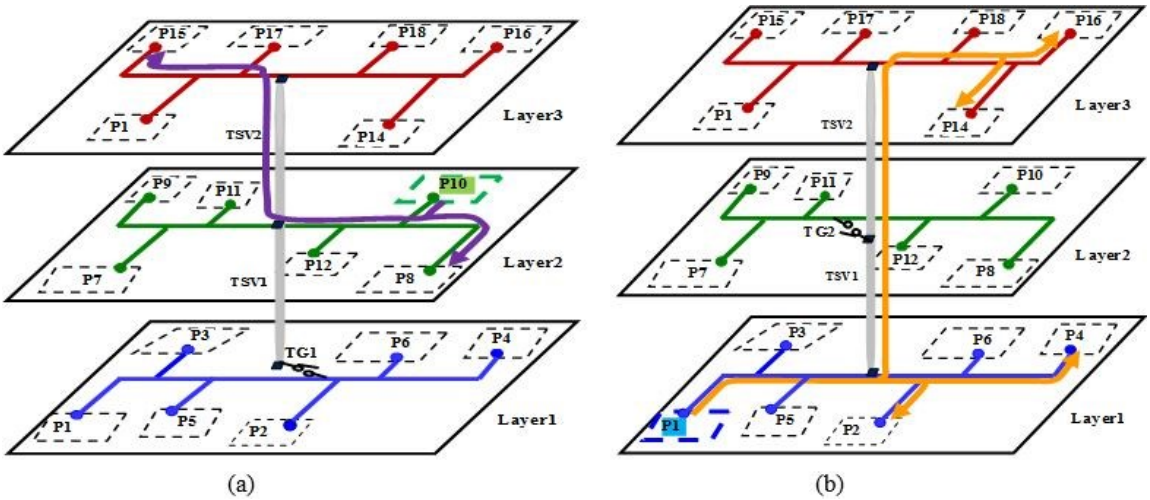


Fig. 6. (a) Sizing the source driver p10 and Embedded bus switches isolate extra local bus capacitive loadings, (a) from source to sink p15 and (b) from source p1 to sink p14 or p16, for reducing the critical access time.

The objective of the *Problem-Case2* is to tune all the source drivers to minimize the critical access time T_{ij} from source i to sink j as possible, that is,

$$\text{Min}(\text{Max}(T_{ij})),\ 1 \le i, j \le m \ \text{ at timing periods of } p \qquad (1)$$

Fig. 7 shows the *Algorithm2* to solve the *Problem-Case2*. The function Find_Crtitcal_AccessTime() is used to find the critical access time from a source to a sink. We can size the source driver located on the critical path to minimize the critical access time. The function RepeaterSizing() is to alternatively size the repeater sizes to maximally reduce the current access time, *reduce_AccessTime*, as possible. If *reduce_AccessTime* is less than *Critical_AccessTime*, the repeater sizes are accepted and the *Critical_AccessTime* is then updated. The above procedure can be repeated until no improvement in critical access time. Finally, we get the critical access time *Tmax* and average access time *Av_Tmax* to a data bus with sized source drivers and embedded bus switches. Expectably, *Tmax* is smaller than *w-Tcri* while *Av_Tmax* is also smaller than *w-Av_Tcri*.

```
Algoritm2: Sized_SourceDriver_Given_BusSwitches()
{   /* A bus topology with the number of m terminals, n segments, and p
       timing periods (p*h is the number of total timings). */
    From Algorithm1 to get the critical access time w-Tcri and average access
    time w-Av_Tcri for a given 3D data bus with embedded bus switches.
    repeat
       Critical_AccessTime = Find_Critical_AccessTime();
       pre_Critical_AcessTime = Critical_AccessTime;
       reduce_AccessTime = MAX;
       For the source driver Bi located on the critical path, let Bi be a unit-size
       bidirectional repeater, s_Bi1 = s_Bi2 = 1;
       repeat
          sizing_AccessTime = RepeaterSizing(s_Bi1);
          if (sizing_AccessTime < reduce_AccessTime)
             reduce_AccessTime = sizing_AccessTime;
          sizing_AccessTime = RepeaterSizing(s_Bi2);
          if (sizing_AccessTime < reduce_AccessTime)
             reduce_AccessTime = sizing_AccessTime;
          if (reduce_AccessTime < Critical_AccessTime)
          {   Critical_AccessTime = reduce_AccessTime;
              Store the source driver Bi including the sizes.
          }
       until no improvement in sizing the source driver Bi
       if (Critical_AccessTime < pre_Critical_AccessTime)
          Store all source drivers including their sizes.
    until Critical_AccessTime ≥ pre_Critical_AccessTime
    Return the critical access time Tmax = Critical_AccessTime and
    average access time Av_Tmax = Total access time / (p*h)
}
```

Fig. 7. Algorithm2 is for access time reduction by sizing source drivers to a given 3D data bus with embedded bus switches.

C. *Algorithm3 for Problem-Case3: Inserted Repeaters for a Given Data Bus with Embedded Bus Switches*

A data bus with embedded bus switches their critical access times and average access time can also be reduced in advance only by inserting signal repeaters into bus wires. Fig. 8 shows that the access time can be decreased by inserting a signal repeater into the middle for a bus wire and sizing their sizes. A bidirectional repeater consists of two opposite unidirectional repeaters B_1 and B_2 in parallel with different sizes s_{B1} and s_{B2}, respectively. The access time from source i to sink j, T_{ij}, based on the Elmore delay model [13] can be derived as follows.

$$T_{ij} = t_{w1} + t_{B1} + t_{w2}, \tag{2}$$

where t_{w1} and t_{w2} are the delays from source i to repeater B_1 and from repeater B_1 to sink j, respectively, and $t_{w1} \propto R_{di}(C_{Li} + C_{w1} + c_B{*}s_{B1}) + R_{w1}(C_{w1}/2 + c_B{*}s_{B1})$ and $t_{w2} \propto r_B/s_{B1}{*}(c_B{*}s_{B2} + C_{w2} + C_{Lj})$. t_{B1} is the intrinsic delay of B_1 and R_{di} is the output resistance of source i and C_{Lj} is the loading capacitance of sink j. R_{w1} and C_{w1} are the respective resistance and capacitance of the bus wire w_1 and C_{w2} is the capacitance of the bus wire w_2.

From (2), increasing the repeater size s_{B1} will reduce the output resistance r_B/s_{B1} as well as reducing the access time. At the same time, the access time will also be increased due to increasing the input capacitance $c_B{*}s_{B1}$. To

truly reduce the access time, we need to carefully size s_{B1} to be the appropriate size. Similarly, the effect of access time is the same for sizing the opposite repeater size s_{B2} of B_2.

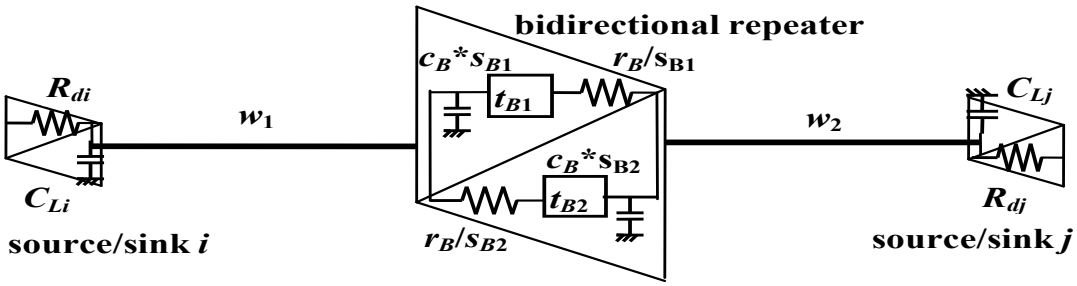


Fig. 8. A bidirectional repeater is inserted into the middle of a bus wire between source/sink terminals i and j.

Fig. 9(a) shows the equivalent Π-RC circuit model of Fig. 9(b) from source i (p11) to sink j (p18) with two embedded bus switches and two inserted bidirectional repeaters. The access time is based on Elmore RC delay model that is the scaled-50% propagation delay. Likely a bus wire, a TSV has also the equivalent RC mode with the resistance $rTSV$ and two half capacitances of $cTSV/2$. The access time T_{ij} from source i to sink j can be formulated as follows (not whole derived due to space limitation).

$$T_{ij} = R_{di}(C_{X1}+c_B*s_{B1}) +...+ t_{B1} + r_B/s_{B1}(c_B*s_{B2}+C_{X2}) +...+ r_G(C_{X3}+c_B*s_{B3}) +...+ t_{B3} + r_B/s_{B3}(c_B*s_{B4}+C_{X4}) +...+ r_{wl} (c_{wl}/2 + C_{Lj}), \quad (3)$$

where C_{X1}, C_{X2}, C_{X3}, and C_{X4} are their lumped capacitances at nodes X1, X2, X3, and X4, respectively.

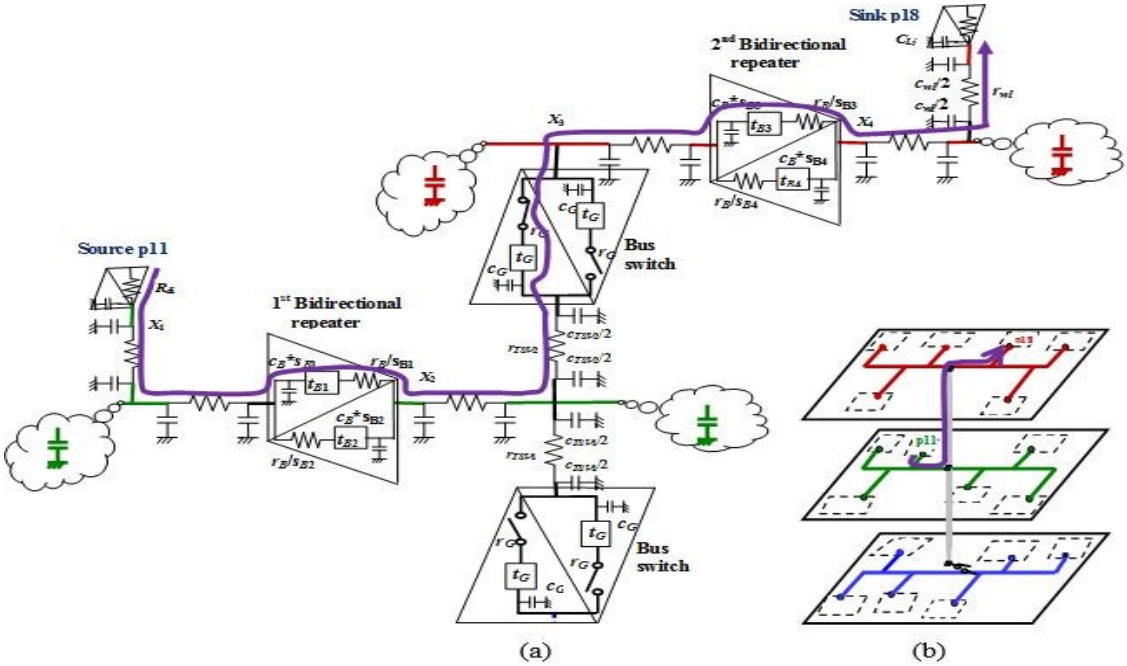


Fig. 9. (a) The equivalent Π-RC circuit model. (b) Data are propagated from source p11 to sink p18 on 3D data bus with embedded bus switches and inserted repeaters.

From (3), we can size two bidirectional repeater sizes to be their appropriate sizes for reducing the access time T_{ij}. The first bidirectional repeater sizes s_{B1} and s_{B2} are alternatively sized until no any improvement for T_{ij}. Then, the second bidirectional repeater sizes s_{B3} and s_{B4} are also alternatively sized until no any improvement for T_{ij}. The sized process for two bidirectional repeaters is repeated until the access time T_{ij} is minimized. Fig. 10 shows the sizing process of access time reduction depending on their appropriate sizes s_{B1}, s_{B2}, s_{B3}, and s_{B4} of repeaters B_1, B_2, B_3, and B_4, respectively.

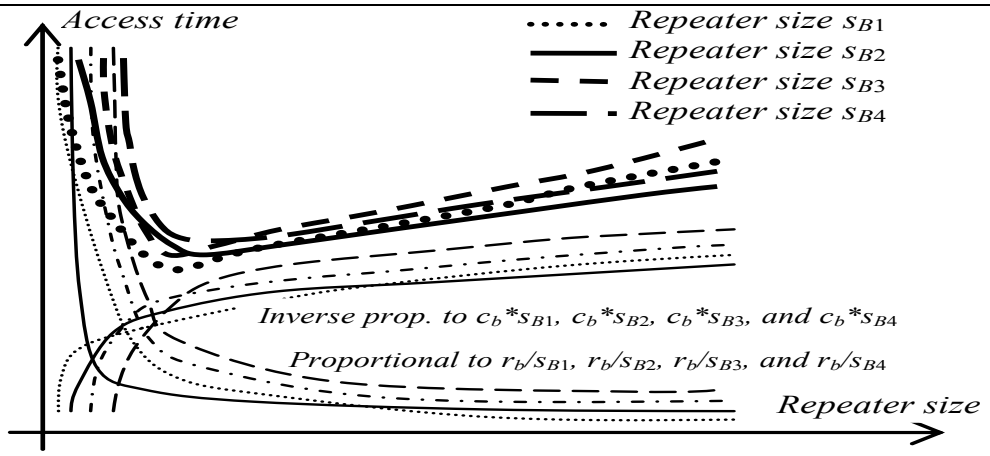


Fig. 10. The sizing process of access time reduction depending on their repeater sizes s_{B1}, s_{B2}, s_{B3}, and s_{B4}.

The objective of the *Problem-Case3* is to insert signal repeaters into all the bus wires to minimize the critical access time, Min(Max(T_{ij})), from source i to sink j as possible, $1 \le i, j \le m$ at timing periods of p. Fig. 11 shows the *Algorithm 3* to solve the *Problem-Case3*. The function Find_Crtitcal_AccessTime() is used to find the critical access time and critical path from a source to a sink. We can insert repeaters into the number of k bus wires located on the critical path and size their repeaters' sizes in the function RepeaterSizing() to minimize the critical access time. The above procedure can be repeated until no improvement in critical access time. Finally, we get the critical access time *Tmax* and average access time *Av_Tmax* to a data bus with inserted repeaters and embedded bus switches. Expectably, *Tmax* is less than *w-Tcri* while *Av_Tmax* is also less than *w-Av_Tcri*.

```
Algorithm3: Inserted_Repeaters_Given _BusSwitches()
{   /* A bus topology with the number of m terminals, n segments, and p
       timing periods (p*h is the number of total timings). */
   From Algorithm1 to get the critical access time w-Tcri and average access
   time w-Av_Tcri for a given 3D data bus with embedded bus switches.
   repeat
      Critical_AccessTime = Find_Critical_AccessTime();
      pre_Critical_AcessTime = Critical_AccessTime;
      reduce_AccessTime = MAX;
      repeat
         For the number of k bus wires along current critical path, let Bi be a
         repeater inserted possibly into a bus wire y and set s_Bi1 = s_Bi2 = 1;
         repeat
            sizing_AccessTime = RepeaterSizing(s_Bi1);
            if (sizing_AccessTime < reduce_AccessTime)
                  reduce_AccessTime = sizing_AccessTime;
            sizing_AccessTime = RepeaterSizing(s_Bi2);
            if (sizing_AccessTime < reduce_AccessTime)
                  reduce_AccessTime = sizing_AccessTime;
            if (reduce_AccessTime < Critical_AccessTime)
            {     Critical_AccessTime = reduce_AccessTime;
                  Store the inserted repeater Bi including location and sizes.
            }
         until no improvement in sizing the repeater Bi
      until no improvement in sizing all the repeaters inserted k bus wires
      if (Critical_AccessTime < pre_Critical_AccessTime)
          Store all inserted repeaters including their locations and sizes.
   until Critical_AccessTime ≥ pre_Critical_AccessTime
   Return the critical access time Tmax = Critical_AccessTime and
   average access time Av_Tmax = Total access time / (p*h)
}
```

Fig. 11. Algorithm3 is for access time reduction by inserting repeaters into bus wires to a given 3D data bus with embedded bus switches.

D. *Algorithm4 for Problem-Case4: Sized Source Drivers and Inserted Repeaters for a Given Data Bus with Embedded Bus Switches*

For a data bus with embedded bus switches, we can combine *Algorithm2* and *Algorithm3* for reducing their critical access time and average access time in advance. The objective of the *Problem-Case4* is to sizing source drivers first and then insert signal repeaters into all the bus wires to minimize the critical access time, Min(Max(T_{ij})), from source i to sink j, $1 \le i, j \le m$ at timing periods of p. Fig. 12 shows the *Algorithm4* for solving the *Problem-Case4*. Finally, we get the critical access time *Tmax* that should be smaller than *w-Tcri* and the average access time *Av_Tmax* that should also be smaller than *w-Av_Tcri*.

```
Algorithm4:Sized_SourceDriver&Inserted_Repeaters_Given_BusSwitches()
{   /* A bus topology with the number of m terminals, n segments, and p
       timing periods (p*h is the number of total timings). */
    From Algorithm1 to get the critical access time w-Tcri and average access
       time w-Av_Tcri for a given 3D data bus with embedded bus switches.
    From Algorithm2 to get the critical access time Tmax and average access
       time Av_Tmax for first sizing source drivers to a given 3D data bus with
       embedded bus switches.
    From Algorithm3 to get the updated critical access time Tmax and updated
       average access time Av_Tmax for then inserting repeaters into bus wires
       to a given 3D data bus with embedded bus switches.
    Return the critical access time Tmax = Critical_AccessTime and
    average access time Av_Tmax = Total access time / (p*h)
}
```

Fig. 12. Algorithm4 is for access time reduction by sizing source drivers first and then inserting repeaters into bus wires to a given 3D data bus with embedded bus switches.

E. *Integrated Algorithm for Data Access Reduction to Problem-Cases1~4 of a 3D Data Bus*

Fig. 13 presents the proposed integrated algorithm, *Integrated_Algorithm*, for data access reduction to the above *Problem-Cases1~4* of a 3D Data Bus.

```
Integrated_Algorithm_for_DataAccessReduction_3D_DataBus()
{   /* A bus topology with the number of m terminals, n segments, and p
       timing periods (p*h is the number of total timings). */
    Do you want to embed bus switches into a 3D data bus?   Yes or Skip
      Yes: perform Algorithm1 to get the critical access times wo-Tcri and w-Tcri
           the average access times wo-Av_Tcri and w-Av_Tcri for a given 3D
           data bus without/with embedded bus switches.
    Do you want to size the sources drivers for a 3D data bus?   Yes or Skip
      Yes: perform Algorithm2 to get the critical access time Tmax and the
           average access time Av_Tmax depending on whether Algorithm1 is
           executed or not.
    Do you want to insert signal repeaters into bus wires and to size?   Yes or Skip
    Yes: perform Algorithm3 to get the critical access time Tmax and the
           average access time Av_Tmax depending on whether Algorithm1 and/or
           Algorithm2 are executed or not.
    Return all the critical access times and average access times for comparison
}
```

Fig. 13. An integrated algorithm is for access time reduction in any mixed cases on a 3D data bus.

The integrated algorithm combines three algorithms, *Algorithm1 to Algorithm3*, and each algorithm gives one of two choices, “Yes” or “Skip”. The integrated algorithm has flexible works to any mixed cases, that is, only works one of three algorithms or combines two or more algorithms. For example, the *Algorithm4* is worked due to the user that chooses three continuously “Yes” for performing *Algorithm1*, *Algorithm2*, and *Algorithm3* for access

time reduction by sizing source drivers first and then inserting repeaters into bus wires to a given data bus with embedded bus switches. Therefore, the comparison in critical access time and average access time for their different mixed cases is very convenient and valuable for selection.

The time complexity of the proposed algorithm is O(M*p*n^4) [5], where M is the maximum repeater size, p is the number of timing periods, and n is the number of bus wires.

IV. Experimental Results

We have implemented the proposed algorithm in C language on an i7 CPU@2.7GHz, dual cores with 8GB RAM, running MS-Windows 10. Table I shows the parameters of 45nm technology [12] based on Elmore RC delay model [13]. Terms r_w and c_w represent the resistance and capacitance of a unit-length wire, respectively. *rTSV* and *cTSV* are the resistance and capacitance of a TSV, respectively. r_B, c_B, and t_B denote the output resistance, input capacitance, and intrinsic delay of a unit-size signal repeater, respectively. r_G, c_G, and t_G introduce the switched-on resistance, input capacitance, and intrinsic delay of a bus switch, respectively.

Table I. Parameters based on 45nm technology

a wire		a TSV		a unit repeater			a bus switch		
r_w	c_w	*rTSV*	*cTSV*	r_B	c_B	t_B	r_G	c_G	t_G
0.1Ω	**0.2fF**	**0.035Ω**	**15.48fF**	**122Ω**	**24fF**	**17ps**	**1Ω**	**1pF**	**10ps**

Since no any benchmarks for 3D data bus, we create nine examples of different bus topologies with three stacked-chip layers for testing our proposed algorithm. Table II shows nine 3D data bus topologies and each data bus has 3 layers and 2 TSVs. *Tlength* is the total length in μm of a 3D data bus. #*Term* and #*Bwire* are the number of terminals as for sources or sinks and the number of bus wires as for the locations of inserted repeaters, respectively. #*Timing* and #*Peri* are the total number of data access timings and timing periods, respectively. #*timing-0-layer*, #*timing-1-layer* and #*timing-2-layer* are the number of data access timings accessed across zero, one, and two layers, respectively. Test0 is the right test case of Fig. 1 as for the explanation going through whole article. For each bus topology, the driving resistance R_d of all the sources and the loading capacitance C_L of all the sinks are assumed to be those of a unit-sized repeater (x1). For a repeater, the size can be sized up to x16.

Table II. Nine examples of different 3D data bus topologies and each bus has 3 layers and 2 TSVs.

Examples	***Tlength***	***#Term***	***#Bwire***	***#Timing /#Peri***	***#timing-0-layer***	***#timing-1-layer***	***#timing-2-layer***
Test0	37702μm	18	38	55/18	24	21	10
CaseF	60461μm	15	29	34/15	20	8	6
CaseG	44069μm	10	20	6/6	6	0	0
CaseG2	44069μm	10	20	16/10	6	4	6
CaseH	42770μm	9	17	6/6	6	0	0
CaseI	58911μm	15	29	34/15	20	8	6
CaseI2	58911μm	15	29	49/15	20	9	20
CaseJ	56006μm	21	39	24/19	24	0	0
CaseK2	69112μm	30	58	100/30	31	44	25

First, we test the case of a 3D data bus whether is embedded bus switches or not. Table III shows the comparison of different 3D data bus topologies without/with embedded bus switches between their TSVs and local buses. In the table, *wo-Tcri* and *w-Tcri* are the critical access times without/with embedded bus switches, respectively. *wo-Av_Tcri* and *w-Av_Tcri* are the average access times without/with embedded bus switches, respectively. The savings *w-Saving* and *w-Av_Saving* in the critical access time and the average access time are reduced up to 54.56% and 60.53% on average, respectively.

Table III. Comparison of 3D data bus without/with inserted bus switches in critical access time and average access time.

Examples	*Tlength*	*#Timing/ #Peri*	*#timing-0-layer*	*#timing-1-layer*	*#timing-2-layer*	*wo-Tcri* (ns)	*wo-Tav* (ns)	*w-Tcri* (ns)	*w-Av_Tcri* (ns)	*w-Saving*	*w-Av_Saving*
Test0	37702µm	55/18	**24**	**21**	**10**	5.014	3.282	2.268	1.376	54.76%	47.11%
CaseF	60461µm	34/15	**20**	**8**	**6**	11.002	6.961	5.366	2.231	51.22%	67.95%
CaseG	44069µm	6/6	**6**	**0**	**0**	7.364	5.502	3.460	1.981	53.01%	63.99%
CaseG2	44069µm	16/10	**6**	**4**	**6**	7.782	5.475	3.673	2.065	52.80%	62.28%
CaseH	42770µm	6/6	**6**	**0**	**0**	7.392	5.781	3.188	2.595	56.86%	55.11%
CaseI	58911µm	34/15	**20**	**8**	**6**	9.834	6.716	4.753	2.361	51.66%	64.84%
CaseI2	58911µm	49/15	**20**	**9**	**20**	9.855	6.842	4.754	2.362	51.76%	65.47%
CaseJ	56006µm	24/19	**24**	**0**	**0**	10.755	5.585	4.734	2.353	55.96%	57.86%
CaseK2	69112µm	100/30	**31**	**44**	**25**	14.729	8.536	5.440	3.398	63.06%	60.19%
Average	-									**54.56%**	**60.53%**

Second, we test the case of sized source drivers for a given 3D data bus with embedded bus switches. Table IV presents the results of critical access time reduction in cases of nine examples. Both *Tmax* and *Av_Tmax* are the maximum access time and average access time, respectively, with embedded bus switches and sized source drivers. *Uloc* and *Loc* are the number of sized source drivers and the total number of source drivers, respectively. *Sizes* is the sum of sized sources drivers' sizes and *Ctime* is the running time. The saving *Saving* in critical access time can be reduced up to 10.43% with the 50% of sized source drivers, the 29 unit sizes of all the sized source drivers, and the running time of 0.169s on average; but the saving of *Av_Saving* in average access time degenerates 0.54%.

Third, we test the case of inserted signal repeaters for a given 3D data bus with embedded bus switches. Table V shows the results of critical access time reduction in cases of all the examples. The savings of *Saving* and *Av_Saving* in the critical access time and average access time are reduced up to 77.99% and 68.15%, respectively, in the running time of 0.747s on average, but pay the 87% of inserted repeaters and 107 unit sizes of all the sized repeaters on average.

Finally, we test the mixed case of sized source drivers and inserted signal repeaters for a given 3D data bus with embedded bus switches. Table VI indicates the results of critical access time reduction in cases of all the examples. The savings of *Saving* and *Av_Saving* in the critical access time and average access time are reduced up to 76.39% and 66.50%, respectively, in the running time of 0.994s on average, but pay the 70% of sized source drivers and inserted repeaters and the 130 unit sizes of all the sized drivers and inserted repeaters on average.

Table IV. Results in case of sized source drivers for 3D data bus with embedded bus switches.

Examples	*Tlength*	*Term*	*#Timin/ #Peri*	*w-Tcri* (ns)	*w-Av_Tcri* (ns)	*Tmax* (ns)	*Av_Tmax* (ns)	*Uloc/Loc*	*Sizes*	*Ctime*	*Saving*	*Av_ Saving*
Test0	37702µm	18	55/18	2.268	1.376	2.074	1.413	14/18	34	0.230s	9.43%	-2.62%
CaseF	60461µm	15	34/15	5.367	2.231	4.862	2.397	9/15	34	0.070s	9.79%	-1.47%
CaseG	44069µm	10	6/6	3.460	1.981	3.033	1.917	5/10	23	0.001s	12.34%	3.25%
CaseG2	44069µm	10	16/10	3.673	2.065	3.265	2.052	5/10	23	0.020s	11.12%	0.67%
CaseH	42770µm	9	6/6	3.189	2.595	2.823	2.549	3/9	17	0.001s	12.19%	1.77%
CaseI	58911µm	15	34/15	4.753	2.361	4.257	2.259	5/15	26	0.080s	10.85%	-1.22%
CaseI2	58911µm	15	49/15	4.754	2.362	4.257	2.379	4/15	24	0.120s	10.85%	-0.69%
CaseJ	56006µm	21	24/19	4.736	2.353	4.312	2.360	6/21	25	0.080s	8.94%	-0.22%
CaseK2	69112µm	30	100/30	5.440	3.398	4.988	3.544	19/30	50	0.920s	8.35%	-4.30%
Average	-							**8/16=50%**	**29**	**0.169s**	**10.43%**	**-0.54%**

Table V. Results in case of inserted signal repeaters for 3D data bus with embedded bus switches.

Examples	*Tlength*	*#Bwire*	*#Timin/ #Peri*	*w-Tcri* (ns)	*w-Av_Tcri* (ns)	*Tmax* (ns)	*Av_Tmax* (ns)	*Uloc/Loc*	*Sizes*	*Ctime*	*Saving*	*Av_ Saving*
Test0	37702µm	38	55/18	2.268	1.376	0.636	0.465	36/38	153	0.940s	71.96%	66.23%
CaseF	60461µm	29	34/15	5.367	2.231	0.902	0.718	26/29	118	0.433s	83.18%	69.59%
CaseG	44069µm	20	6/6	3.460	1.981	0.815	0.761	19/20	69	0.020s	76.44%	61.58%
CaseG2	44069µm	20	16/10	3.673	2.065	0.889	0.743	18/20	76	0.080s	75.78%	63.99%
CaseH	42770µm	17	6/6	3.189	2.595	0.941	0.910	15/17	56	0.030s	70.47%	64.92%
CaseI	58911µm	29	34/15	4.753	2.361	0.900	0.724	23/29	97	0.209s	83.06%	67.53%
CaseI2	58911µm	29	49/15	4.754	2.362	0.901	0.719	25/29	116	0.450s	81.05%	69.56%
CaseJ	56006µm	39	24/19	4.736	2.353	0.972	0.649	26/39	95	0.493s	79.46%	72.42%
CaseK2	69112µm	58	100/30	5.440	3.398	1.057	0.762	48/58	185	4.072s	80.57%	77.56%
Average	-							**27/31=87%**	**107**	**0.747s**	**77.99%**	**68.15%**

From the above experimental results, we have their discussions for different mixed cases as follows. For the Case1 that bus switches are embedded into a given 3D data bus, their access time reduction is very effective. For the Case2 that sizing source drivers only is to a given 3D data bus with embedded bus switches, their access time minimization in advance is very limited. Moreover, for the Case3 that inserting signal repeaters into bus wires to a given 3D data bus with embedded bus switches, the maximum reduction for their access time in advance is obviously improved and can be a good strategy. For the Case4 that mixing sized source drivers and inserted repeaters is to a given 3D data bus with embedded bus switches, their data access time reduction in advance is slightly less than Case3.

Table VI. Results in case of sized source drivers and inserted signal repeaters for 3D data bus with embedded bus switches.

Examples	*Tlength*	*#Bwire*	*#Timing/ #Peri*	*w-Tcri* (ns)	*w-Av_Tcri* (ns)	*Tmax* (ns)	*Av_Tmax* (ns)	*Uloc/Loc*	*Sizes*	*Ctime*	*Saving*	*Av_ Saving*
Test0	37702µm	38	55/18	2.268	1.376	0.742	0.528	45/(18+38)	148	1.380s	67.27%	61.61%
CaseF	60461µm	29	34/15	5.367	2.231	0.938	0.747	35/(15+29)	154	0.761s	82.52%	68.37%
CaseG	44069µm	20	6/6	3.460	1.981	0.930	0.851	17/(10+20)	71	0.040s	73.13%	57.04%
CaseG2	44069µm	20	16/10	3.673	2.065	0.924	0.768	24/(10+20)	104	0.119s	74.84%	62.81%
CaseH	42770µm	17	6/6	3.189	2.595	0.941	0.906	18/(9+17)	73	0.029s	70.47%	65.10%
CaseI	58911µm	29	34/15	4.753	2.361	0.922	0.740	29/(15+29)	129	0.320s	80.59%	66.83%
CaseI2	58911µm	29	49/15	4.754	2.362	0.924	0.736	30/(15+29)	145	0.530s	80.56%	68.83%
CaseJ	56006µm	39	24/19	4.736	2.353	1.021	0.676	31/(21+39)	115	0.680s	78.43%	71.27%
CaseK2	69112µm	58	100/30	5.440	3.398	1.101	0.804	64/(30+58)	231	5.770s	79.77%	76.34%
Average	-							**33/47=70%**	**130**	**0.994s**	**76.39%**	**66.50%**

Fig. 14 shows the 3D data bus topology of Case K2. The critical access time is 14.729 ns from source p22 to sink p28 for the data bus without embedded bus switches and inserted repeaters. Based on inserted signal repeaters for the data bus with embedded bus switches, the critical access time is reduced to be 1.057 ns. In the figures, two numbers located on the middle of a bus wire are the sizes of an inserted bidirectional repeater whereas only one number is the size of an inserted unidirectional repeater.

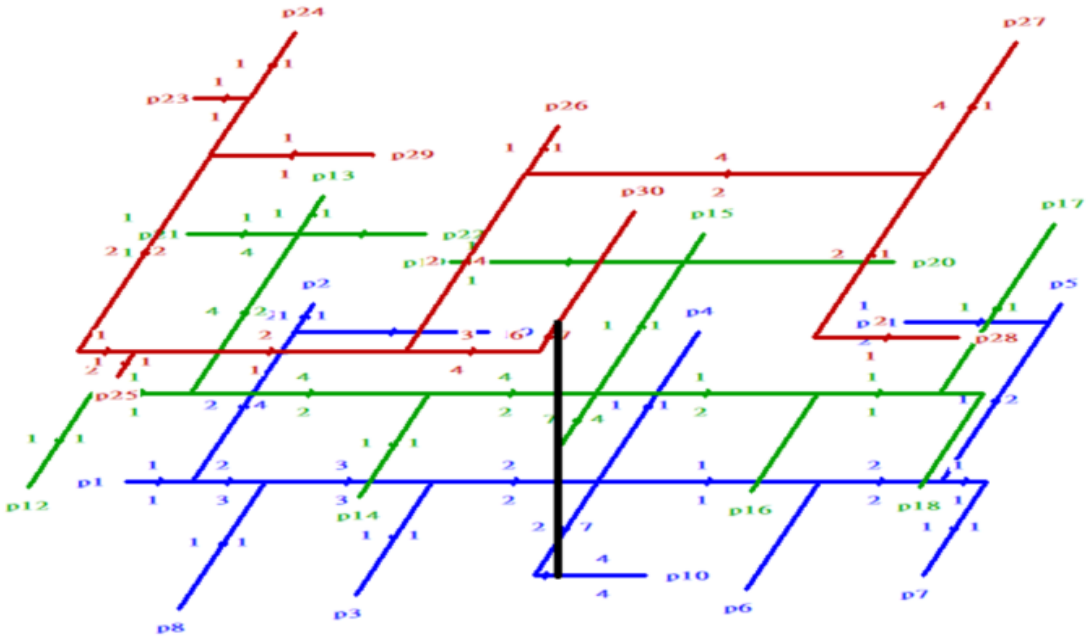


Fig. 14. The bus topology of CaseK2 is based on inserted repeaters to a given 3D data bus with embedded switches.

V. CONCLUSION

The proposed algorithm based on embedded bus switches and inserted signal repeaters has successfully applied to the 3D stacked-layer data bus topology and can significantly reduce the critical access time. For the higher stacked-layer 3D data bus, investigating a super-speed bus switch and considering more practical TSV model [14] can be extended as our future work to avoid any effects in the minimization of critical access time.

ACKNOWLEDGMENT

The author would like to thank for financial support from NHU-107 research project.

REFERENCES

[1] EE Times, *The state of the art in 3D IC technologies*, Nov. 27, 2013.
[2] K. Cho, H-S Na, T.W. Cho, and Y. You, "Analysis of system bus on SoC platform using TSV interconnection," in *Proc. The 4th ASQED*, 2012, pp. 255-259.
[3] M. Daneshtalab, M. Ebrahimi, and J. Plosila, "HIBS- Novel Inter-layer Bus Structure for Stacked Architectures," in *Proc. IEEE International Conf. on 3D System Integration*, 2012, pp. 1-7.
[4] I.G. Thakkar and S. Pasricha, "3D-Wiz: A novel high bandwidth, optically interfaced 3D DRAM architecture with reduced random access time," in *Proc. IEEE International Conf. on Computer Design*, 2014, pp. 1-7.
[5] C-C Tsai, "Repeater Insertion for 3D Data Bus with TSVs for Reducing Critical Propagation Delay," in *Proc. International Conference on Computer Science and Information Engineering*, 2015, pp. 203-208.
[6] C-C Tsai, "A Heuristic Algorithm for Critical Access Time Minimization on 3D Data Bus," *The 27th VLSI Design/CAD Symposium*, Paper ID:182, Aug. 2016.
[7] C-C Tsai, "An Effective Algorithm for Minimizing the Critical Access Time of a 3D-Chip Data Bus," *International Journal of Electronics Communication and Computer Engineering*, Vol. 9, No. 4, July 2018, pp. 117-123.
[8] C-C Tsai, "Minimizing Critical Access Time for 3D Data Bus Based on Inserted Bus Switches and Repeaters," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2017, pp. 140-145.
[9] C-C Tsai, "Embedded Bus Switches on 3D Data Bus for Critical Access Time Reduction," *The 9th IEEE Latin American Symposium on Circuits and Systems*, Paper ID-16, Feb. 2018.
[10] IDTQS3245 data sheet, IDT Co., Nov. 2014.
[11] 74VHCT126AFT data sheet, Toshiba Co., Nov. 2014.
[12] http://ptm.asu.edu/.
[13] W.C. Elmore, "The transient response of damped linear networks," *Journal of Applied Physics*, Vol. 19, pp. 55-63, 1948.
[14] T. Bandyopadhyay, K-J Han, D. Chung, R. Chatterjee, M. Swaminathan, and R. Tummala, "Rigorous electrical modeling of through silicon vias with MOS capacitance effects," *IEEE Trans. Components, Packaging, and Manufacturing Technology*, Vol. 1, No. 6, Jun 2011, pp. 893-903.

AUTHORS' PROFILE

Chia-Chun Tsai received his B.S. degree in Industrial Education from National Taiwan Normal University, Taipei, Taiwan, ROC, in 1978, and both M.S. and Ph.D. degrees in Electrical Engineering from National Taiwan University, Taipei, Taiwan, ROC, in 1987 and 1991, respectively. From 1978 to 1989, he was a specialist teacher of electronic maintenance at Taiwan Provincial Hsin-Hua and Taipei Municipal Nan-Kang Technology High Schools, respectively. From 1989 to 2005, he served at the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan, ROC. From 1994 to 1995 and 2000 to 2001, he was on leave from National Taipei University of Technology and worked on postdoctoral research at University of California, San Diego, and North Carolina State Univer-
-sity, respectively. Since 2005 he has been with the Department of Computer Science and Information Engineering, Nanhua University, Chiayi, Taiwan, ROC, where he is a Full Professor. His current research interests include VLSI design automation and mixed-signal IC designs.