Abstract - As the processing capabilities and operating frequency of embedded system are growing, so is the needed data bandwidth to fully utilize the processing capability. The ability to transfer huge amount of data between the embedded core and external devices is required for efficient system operation. In this paper, the data communication architecture for the mixed-clock system is proposed. The dynamic priority adaptation algorithm for bus arbitration is proposed for bandwidth guarantee. The communication architecture that incorporates the proposed arbitration algorithm adapts the priority of communication components dynamically based on the information from FIFO. The experiments show that the measured bandwidth of each component traces the required bandwidth well compared to the other arbitration algorithms.

I. Introduction

With the advent of system-on-a-chip (SOC) time-to-market pressure and the requirement of many components for the system functionality made it difficult for a single design team to design the whole system. Therefore, design reuse, or use of IP’s (intellectual Properties) has become a nearly mandatory design methodology for the design of electronic systems including embedded systems. The most important requirement of the embedded systems is performance and cost efficiency. As the embedded systems incorporate a larger number of components the relative importance of each communication component in the mixed-clock embedded system. The novel arbitration mechanism, dynamic priority adaptation (DPA) is introduced. The proposed communication architecture incorporates the enhanced DMA architecture with dynamic priority adaptation. In section II, the communication architecture for the mixed-clock system is shown and limitations are explained. In section III, the communication architecture with the proposed dynamic priority adaptation algorithm is proposed. The experiments are shown in section IV and the conclusion is in section V.

II. The Communication Architecture for the Mixed-Clock System

The data transfer between communication components that have difference clock frequencies is usually implemented by First-In First-Out (FIFO) hardware. DMA is a technique by which blocks
of data can be transferred from one part of the memory or peripherals to the other part without control of the embedded core. The DMA controller is usually used to transfer data between several FIFO’s and the memory of communication component. The conventional data communication architecture for several communication components with DMA controller is shown in Fig. 1.

The clock of DMA is synchronized to the parent communication component in Fig. 1 and moves data for one of the FIFO’s. Therefore, the arbitration mechanism that determines priority of FIFO’s is important for the high-performance data communication in the mixed-clock system. It is required to design the efficient arbiter that selects the winner FIFO for DMA transfer.

![Fig. 1. The mixed-clock data communication system. The DMA controller moves data between FIFO’s of several communication components and the parent communication component.](image)

A. Issues

The important issue of the arbiter in the communication architecture is minimization of overall execution time. The scheduling algorithms proposed in [1] and [2] determine the sequence of the execution of each communication component at the compile time. This static scheduling algorithm is not applicable to the general embedded system where the state of each communication component is determined dynamically by external environment. The dynamic behavior requires the arbiter to select the winner for the data communication based on current status of each communication component. Therefore, it is required for the arbiter of the general embedded system to guarantee the bandwidth of each communication component that changes dynamically.

In the mixed-clock system where FIFO is used, it is important to check the FIFO status to guarantee the bandwidth of the communication component. The importance of the FIFO status is shown in Fig. 2. The case (a) shows DMA service order and data transfer status of each FIFO when FIFO depth is unlimited. The overall execution time is 10 time slots. The case (b) shows the data transfer when FIFO depth is limited to one time slot. It means that the FIFO becomes full or empty when the data transfer occurs for one time slot between the FIFO and the communication component. Although the overall execution time is the same for case (a) and case (b), bandwidth requirement is not satisfied in case (b). If the FIFO channel 2 is connected to the communication component such as image generator for MPEG system, the frame rate of case (b) decreases to 1/3 of that of the case (a).

![Fig. 2. The importance of the FIFO status in case of (a) when FIFO depth is unlimited and the communication component always succeeds to transfer data with FIFO and (b) when FIFO depth is limited and the communication component fails to transfer data when FIFO is full or empty.](image)

B. Limitation of Conventional Arbitrations

There are several conventional arbitration mechanisms such as round-robin, fixed priority, and time division multiplexed access. The data transfer when round-robin mechanism is used was shown before in Fig. 2.

The data transfer with fixed priority arbitration mechanism is shown in Fig. 3. The order of priorities for each FIFO channel is 2→1→0 because FIFO channel 2 requires the highest bandwidth in the example. The bandwidth requirement is satisfied well for FIFO channel 2, but the FIFO channel 0 can be starved because the channels with highest priority monopolizes the shared bus.

![Fig. 3. The data transfer when the fixed priority arbitration mechanism is used. The service for FIFO channel 0 can be starved.](image)

III. Communication Architecture with Dynamic Priority Adaptation

We designed a communication architecture that incorporates DMA architecture with novel arbitration mechanism, dynamic priority adaptation. It arbitrates between a multitude of FIFO’s and the parent communication component and determines the transfer parameters by itself. The child FIFO channels are usually the component that comprises the embedded core such as peripheral subsystems and custom ASIC cores operating at different frequencies. For the mixed-clock operation between the child
channels and the parent channel, FIFO’S are used for burst data transfer. The FIFO for the DPA algorithm has additional signals to report the current status of FIFO to the DMA controller. In this section, we describe the proposed dynamic priority adaptation (DPA) mechanism for FIFO arbitration.

A. The Architecture

The block diagram of the proposed communication architecture is shown in Fig. 4 with several communication components (CC). The DPA arbiter requests current status of each FIFO at each clock cycle. The selected FIFO channel by DPA arbiter sends the information about internal status of FIFO to FIB. DPA arbiter gathers the information and selects one of them based on dynamic priority adaptation algorithm. The winner candidate information is sent to DMA controller.

The burst length calculator (BLC) computes current burst length of the winner candidate based on the FIFO information of the selected winner candidate.

The DMA controller selects the winner candidate as current winner if the processed data transfer is finished. If the winner candidate is determined, the new data transfer is started with the burst length computed by BLC.

B. Dynamic Priority Adaptation

We have defined the current status of FIFO as follows:

- \( d_i \): The depth of \( i \)-th FIFO. It is the same to the count of buffer cells inside of the FIFO.
- \( c_{i,t} \): The current fill count of \( i \)-th FIFO at time \( t \). It is the occupied spaces in the FIFO. Time \( t \) is the clock cycle count.
- \( s_{i,t} \): The status sampling period. It is the clock cycle count of the interval of checking the current status of \( i \)-th FIFO.
- \( v_{i,t} \): The fill speed of \( i \)-th FIFO. It is the rate of filling of FIFO at clock instant \( t \). The fill speed is defined as
  \[
  v_{i,t} = \frac{d_{i,t} - c_{i,t-1}}{s_{i,t}}
  \]

The fill speed is initialized at each sampling point.

- \( f_{i,t} \): The fail count of \( i \)-th FIFO at time \( t \). When the FIFO becomes full or empty, the communication component fails to transfer data to/from the FIFO when it tries to transfer data. The fail count of FIFO, \( f_{i,t} \), is defined as the fail count of communication component for the \( i \)-th FIFO during \( s_{i,t} \). It is reset at the sampling point.

The priority of \( i \)-th communication component is determined by the following heuristic equation,

\[
p_{i,t} = c_{i,t} + (c_{i,t} + v_{i,t}d_i)f_{i,t}s_{i,t}.
\]

The priority is proportional to the sum of the current fill count, \( c_{i,t} \) and the fill speed, \( v_{i,t} \). The fill speed is the gradient of the fill count at each sampling point. If fill speed for \( i \)-th component is larger, it means that the communication component is likely to fail to transfer the data in the near future. The arbitration algorithm should increase the priority for that component to trace the required bandwidth. In other words, the current gradient field is required to estimate the increment or decrement of the bandwidth of \( i \)-th communication component. Therefore, the priority of \( i \)-th FIFO should be proportional to the current fill count and the current gradient of fill count.

The priority of the communication component that failed to transfer data with the FIFO is increased by \( f_{i,t} \) that represents the frequency of data transfer failure. When \( f_{i,t} \) equals to 0 at the beginning of the system operation, the current fill count \( c_{i,t} \) becomes the priority value.

To decrease the hardware complexity, the equation can be modified as follows if \( s_{i,t} \ll d_i \) (The sampling period in cycle count \( \ll \) The FIFO depth).

\[
p_{i,t} \equiv c_{i,t}f_{i,t}s_{i,t} + (\Lambda c_{i,t}f_{i,t})d_i = c_{i,t}f_{i,t}(s_{i,t} + d_i) - c_{i,t-1}f_{i,t}d_i \equiv c_{i,t}f_{i,t}(s_{i,t} + d_i) - c_{i,t-1}f_{i,t-1}(s_{i,t} + d_i)
\]

![Fig. 4. The proposed communication architecture. The FIFO information bus (FIB) carries the information about FIFO’s. DPA arbiter and BLC compute the winner candidate and the burst length based on FIB.](image_url)

The term \( (s_{i,t} + d_i) \) is constant value for \( i \)-th FIFO. Therefore, \( p_{i,t} \) can be computed using two multipliers for each FIFO. The decision of sampling period depends on the average variance of the required bandwidth. The large variance means that the bandwidth of each FIFO changes frequently. The sampling period should be small so that DPA arbiter can adapt to the change of bandwidth. The small sampling period does not show good adaptation at all times. If the sampling period is so small, the fill speed shows the instant behavior. DPA arbiter can not consider the gradual behavior of required bandwidths in this case.

The burst length is also computed based on the FIFO information. The heuristic equation for the burst length computation is as follows.

\[
b_{i,t} = \frac{c_{i,t} + s_{i,t}}{d_i}
\]

We can use the fill count of FIFO as the current burst length but it is not preferred because it can starve the other FIFO’s. Therefore, the fill speed to the FIFO depth is used to scale the current fill count. The burst length becomes smaller than the current fill count.

This equation can be modified as follows to remove the division operation.

\[
b_{i,t} = (c_{i,t} \gg \Lambda)(1 + (\Lambda c_{i,t} \gg n_i)) \quad \text{where } n_i = \log_2 d_i
\]
IV. Experiments

To demonstrate the excellence of the proposed DPA arbitration mechanism, we have implemented the simulator for the mixed-clock data communication. We have simulated several communication components with four arbitration mechanisms.

**ARBIT 0** : Round-Robin arbitration. The fill count for the selected arbitration candidate is used as the burst length.

**ARBIT 1** : Fixed-Priority arbitration. The fill count is used as the burst length.

**ARBIT 2** : Dynamic priority arbitration based on the current fill count that selects the FIFO channel with the maximum fill count as the arbitration winner. The fill count is used as the burst length.

**ARBIT 3** : DPA arbitration.

The example communication architecture has four communication components (CC). The required bandwidth of communication components varies randomly. The bandwidth of DMA controller is 528Mbytes/sec. The sampling period $s_i$ is 32 clock cycles.

The DMA controller starts the transaction as soon as possible the DPA arbiter and BLC finishes the selection of the arbitration winner and the computation of the burst length. The overhead for the arbitration is not large because the arbitration operation is executed with the data transfer in parallel.

The average bandwidth of each component is shown in Table 1.

<table>
<thead>
<tr>
<th>CC #</th>
<th>Required average bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0</td>
<td>33 data transfers/1500ns</td>
</tr>
<tr>
<td>#1</td>
<td>50 data transfers/3750ns</td>
</tr>
<tr>
<td>#2</td>
<td>100 data transfers/6000ns</td>
</tr>
<tr>
<td>#3</td>
<td>63 data transfers/4500ns</td>
</tr>
</tbody>
</table>

The graphs for the required bandwidth and the measured bandwidth of the communication component #1 are shown in Fig. 5. The X axis is time in µs and the Y axis is the bandwidth in data count/µs. The required bandwidth is drawn as the solid line and the measured bandwidth is in dotted line.

The measured bandwidth between 80ms and 120ms does not trace the required bandwidth well in case of (a), (b) and (c). The measured bandwidth fluctuates because the DMA controller is in service for the other interfaces and does not serve for interface #1. The trace of case (d), the proposed DPA algorithm traces the required bandwidth well. The consideration of the internal status of FIFO’s and burst length control guarantees the required bandwidth of each communication components.

The bandwidth traces for CC #2 is shown in Fig. 6.
Fig. 6. The trace of the measured bandwidth of CC #2

The fluctuation level of the required bandwidth of CC #2 is not high compared to that of CC #1. The ARBIT 0 shows good performance but ARBIT 1 and ARBIT 2 is inferior to trace the required bandwidth. The DPA algorithm also shows good performance in case that the fluctuation level is low. The dynamic bandwidth trace for CC #0 and CC #3 is good in case of DPA algorithm (It is not shown here).

MSE (Mean Square Error) is the measure of control and quality and is used as the level of the quality of bandwidth guarantee. MSE equals the mean of the squares of the deviations from target. In this experiment, MSE is the average of square of bandwidth difference at each sampling point. MSE of bandwidth for each component is shown in Table 2. The percentage of MSE compared to ARBIT0 is shown in parenthesis. For each experiment, average bandwidth and bandwidth variance are specified in unit of 1 data transfer/100ns. The cycle count of each experiment is 120000 cycles.

MSE of DPA arbitration algorithm is small compared to the other arbitration algorithms. Although MSE of ARBIT1 or ARBIT2 is larger than that of ARBIT0 for CC1 in experiment 1, CC0 for experiment 2 and several cases, MSE of DPA shows improvement for all cases except CC2 in experiment 3. In experiment 3, variances for CC0 and CC2 are large and the summation of average bandwidth for each component almost equals to the bandwidth of DMA controller. DPA algorithm is inferior in this case.

Table 2. MSE of bandwidth for each component.

<table>
<thead>
<tr>
<th>Component</th>
<th>CC0</th>
<th>CC1</th>
<th>CC2</th>
<th>CC3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW (1/100ns), Variance (1/100ns)</td>
<td>2.2, 1.4</td>
<td>1.3, 0.8</td>
<td>1.7, 0.1</td>
<td>1.4, 0.5</td>
</tr>
<tr>
<td>ARBIT0</td>
<td>695.1 (100%)</td>
<td>423.1 (100%)</td>
<td>645.3 (100%)</td>
<td>81.1 (100%)</td>
</tr>
<tr>
<td>ARBIT1</td>
<td>694.2 (99.8%)</td>
<td>425.1 (100.6%)</td>
<td>644.3 (99.8%)</td>
<td>83.1 (115%)</td>
</tr>
<tr>
<td>ARBIT2</td>
<td>681.3 (98.0%)</td>
<td>378.6 (89.7%)</td>
<td>648.9 (100.5%)</td>
<td>79.8 (94.7%)</td>
</tr>
<tr>
<td>DPA</td>
<td>602.5 (86.6%)</td>
<td>367.2 (86.9%)</td>
<td>593.8 (92.0%)</td>
<td>38.2 (47.1%)</td>
</tr>
<tr>
<td>Experiment 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BW (1/100ns), Variance (1/100ns)</td>
<td>3.2, 3.3</td>
<td>1.9, 0.9</td>
<td>2.0, 1.1</td>
<td>1.6, 0.5</td>
</tr>
<tr>
<td>ARBIT0</td>
<td>6094.2 (100%)</td>
<td>447.8 (100%)</td>
<td>516.0 (100%)</td>
<td>34.2 (100%)</td>
</tr>
<tr>
<td>ARBIT1</td>
<td>6180.8 (101%)</td>
<td>447.8 (100%)</td>
<td>519.2 (100.6%)</td>
<td>32.9 (96.2%)</td>
</tr>
<tr>
<td>ARBIT2</td>
<td>5085.3 (83.4%)</td>
<td>443.5 (99.0%)</td>
<td>451.9 (87.6%)</td>
<td>36.6 (107.0%)</td>
</tr>
<tr>
<td>DPA</td>
<td>5140.7 (84.3%)</td>
<td>335.0 (74.8%)</td>
<td>429.7 (83.3%)</td>
<td>20.4 (59.6%)</td>
</tr>
<tr>
<td>Experiment 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BW (1/100ns), Variance (1/100ns)</td>
<td>2.2, 1.4</td>
<td>1.9, 2.2</td>
<td>2.0, 1.4</td>
<td>1.6, 1.0</td>
</tr>
<tr>
<td>ARBIT0</td>
<td>263.9 (100%)</td>
<td>384.0 (100%)</td>
<td>1414.0 (100%)</td>
<td>769.3 (100%)</td>
</tr>
<tr>
<td>ARBIT1</td>
<td>263.4 (99.8%)</td>
<td>392.6 (102.2%)</td>
<td>1395.1 (98.6%)</td>
<td>772.9 (100.4%)</td>
</tr>
<tr>
<td>ARBIT2</td>
<td>212.3 (80.4%)</td>
<td>405.3 (105.5%)</td>
<td>1343.2 (94.9%)</td>
<td>205.8 (26.8%)</td>
</tr>
<tr>
<td>DPA</td>
<td>152.3 (57.7%)</td>
<td>44.3 (1.1%)</td>
<td>1447.7 (102.4%)</td>
<td>191.2 (24.9%)</td>
</tr>
</tbody>
</table>

V. Conclusion

We have proposed a communication architecture that enables the efficient data communication in the mixed-clock system. The
The proposed DPA algorithm arbitrates between communication components in the proposed communication architecture. DPA algorithm adapts the priority of communication component and traces the dynamic behavior of bandwidth of each communication component. The graph of the measured bandwidth in example system with DPA shows that the bandwidth using DPA traces the required bandwidth very well. The measured MSE for DPA is about 70% of that of the other arbitration algorithms.

VI. References


