Extended Large (3-D) Integration Technology

Seventh Framework Programme
FP7-ICT-2007-1
Project Number: FP7-ICT-215030

D1.4: System Concept Simulation

Version 1.0
Extended Large (3-D) Integration TEchnology

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Extended Large (3-D) Integration Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Number</td>
<td>ELITE-215030</td>
</tr>
<tr>
<td>Document Title</td>
<td>System Concept Simulation</td>
</tr>
<tr>
<td>Work Package</td>
<td>WP2</td>
</tr>
<tr>
<td>Dissemination Level</td>
<td>Public (PU)</td>
</tr>
<tr>
<td>Lead Beneficiary</td>
<td>Royal Institute of Technology - KTH</td>
</tr>
<tr>
<td>Document Editors</td>
<td>Awet Yemane, Zhonghai Lu, Hannu Tenhunen</td>
</tr>
<tr>
<td>Version</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table of contents

1. Introduction ......................................................................................................................... 4
2. The ELITE Conceptual Model ............................................................................................ 4
2.1. NAND flash ........................................................................................................................ 4
2.2. TSV ..................................................................................................................................... 5
2.3. NOC ................................................................................................................................... 5
2.4. The ELITE model ................................................................................................................ 5
3. System Simulation .............................................................................................................. 7
3.1. Setup ................................................................................................................................... 7
3.2. Results ................................................................................................................................. 7
3.2.1. Average raw latency ......................................................................................................... 7
3.2.2. Average normalized latency ............................................................................................ 8
3.2.3. Average throughput ......................................................................................................... 9
3.2.4. Area .................................................................................................................................. 9
4. Conclusion ........................................................................................................................ 10
5. References ......................................................................................................................... 10
1. Introduction

Solid State Drives – SSD - based on NAND flash technology have shown a sharp growth in the market. It is believed that the SSD eventually replaces the mechanical hard disk drives - (HDDs), when the memory densities become comparable. The main goal of ELITE project is to realize low cost, high stacking scalable memory devices with increasing performance to be used as SSDs. Setting this goal requires a conceptual design, modelling and simulation which is potentially and possibly realizable taking into account all aspects including the cost, scalability and performance. The conceptual design helps to reduce the design cycle time and design and manufacturing cost. Also it reduces risk of using leading-edge technologies [1].

This report discusses high level flash memory organization and architectural modelling and simulation based on ELITE 3D integration technology specification. The target is to measure the performance of a conceptual model that integrates the features of next generation technologies and design techniques; the flash memory, 3D stacking and Network-on-Chip (NoC) communication technique.

2. The ELITE conceptual model

2.1. NAND Flash

There is a sharp market growth of NAND flash memories. The growth can be attributed mainly to two factors. Firstly, the expansion of the use of mobile devices such as PDAs, cameras, phones and computers has kept the demand of NAND flash memories all time high and with increasing trend. These consumer-based products led to the mass production of flash memory devices and kept the interest of industries higher on inversing in additional research of the devices.

The second factor influencing the growth of NAND flash memory is that the overall cost per bit of NAND flash is low and has faster erasure speed compared to the NOR based or other counterparts. The erasure speed is important because it is done for each data write process. The Data-write is made by programming an array of floating-gate transistors. This is performed first by erasing then followed by setting bits for an array of memory cells simultaneously. The basis of for the conceptual model is the Quimonda flash memory shown below in Figure 1 which has a page data-write time of 200 us with page size of 4 KB and density of 8 Gbits in 150 mm² area in 48 nm technology [2].

![Figure 1: Quimonda’s 8 Gbits NAND flash memory](image-url)
2.2. TSV

The ELITE 3D stacking uses TSV as vertical wires. They are primarily used as wires for vertical signaling while at the same time as a mechanical support strengthening the 3D structure. TSVs can be treated as local wires reasonably because like local wires, TSV length gets shorter with scaling. Moreover a TSV driven by a device, the TSV RC effect is insignificant and the overall device resistance is the dominant factor [3, 4].

2.3. NoC

Networks-On-Chip (NoC) – is the means of communication in complex System-On-Chip (SoC) platforms where several resources, such as processors, memory blocks designed as IP Cores with specific functionalities are integrated. Data from one resource to the other are transmitted as packets and the traffic routing decisions in the network and switching of packets is done in the switches. The other alternative for communication is to use point to point connection which increases the complexity of interconnects or a bus, which manages its traffic path through ‘time slices’ shared among the resources. But with increasing number of resources, the bus becomes inefficient. Hence NoC based communication emerges as the efficient solution. The performance of a functional NoC is fully exploited specially in a multi-processor environment where scalable and parallel processes are common. In this case, multiple processors have the choice to send and receive packets from any one of the resources available in the network [5].

![Figure 2: 3D Flash Memory NOC model](image)

2.4. The ELITE model

The model considers a multi-layer, multi-processor distributed memory system with high density flash banks stacked across a 3D structure with NoC as a means of communication. The model seen in Figure 2 has 17 layers (L0-L16). On the ground layer (L0), 16 microcontrollers are placed, each connected to a NoC switch. On each of the next 16 layers (L1 – L16), there are 16 banks of flash memory. The 16 layers memory stacking was envisioned in ELITE deliverable D1.1 [6]. The NoC structure has 4x4x17 switches which are connected horizontally via planar interconnect wires and vertically via TSVs. Each switch is connected to a resource (i.e., either to Flash banks or to microcontrollers).
The microcontrollers make read/write requests to a specific address in the memory banks. These requests are packetized and passed to the NoC switch. The packetization process involves setting the destination address and adjusting the data into a specific payload width. This is done in the Network Interface (NI) block as shown in Figure 4. A relative addressing scheme is used to send a packet from the source to the destination.

In the NoC switch the request is encapsulated as packets with their own destination address. These destination addresses are the specific memory banks connected to the switches in the next top layers. Upon arrival of the packets to their destination address, they are unwrapped and decoded in the NI. According to the request type, the memory bank circuitries do a specific function of read or write and send back an acknowledgement to the sender microcontroller in the ground layer.

![Diagram of NoC switches and wires](image)

Figure 3: A packet traverse NoC switches and wires to reach the destination.

The flash banks are divided into blocks, each with a set of pages. The write operation involves the physical programming of the flash blocks. A block or a page in a block can be programmed. The process starts by sending a sequence of commands and addresses followed by data from the microcontroller to the memory blocks. Upon successful completion of programming, the banks send back a confirmation signal as acknowledgement to the microcontroller. This whole write operation takes place in a period, write time, $T_{\text{write}}$, expressed as follows.

$$T_{\text{write}} = T_{\text{setup}} + T_{\text{nwk}} + T_{\text{trans}} + T_{\text{prog}}$$

Where $T_{\text{setup}}$ – the setup time to prepare the sequence in the microcontroller

$T_{\text{nwk}}$ – the round trip network delay in the wires and switches.

$T_{\text{trans}}$ – the address translation time taken in the flash translator.

$T_{\text{prog}}$ – the physical erase and programming of the floating gates

![Diagram of Network Interface (NI) on both ends of the network](image)

Figure 4: The Network Interface – NI – on both ends of the network
The microcontrollers do the command and address sequence preparation and send it to their NI for packetization. The sequences, in the form of packets, hop from a switch to another in every other clock cycle. The network delay is measured in terms of hops. The whole hoping path shown in Figure 3 involves at least a microcontroller, one or more switches and a flash bank. The address translation takes place in the NI of the flash bank after depacketization. Then according to the sequence, the specific command is executed in the specific address, i.e., in this case the page programming is performed.

3. System Simulation

The simulation is done on 3D ELITE NOC VHDL model which has the following features [8].

- Hot-potato – buffer-less
- Non-minimal – load dependant deflection routing
- Packet switching - non reserved, per-hop based setup
- Connectionless - no prior path is established
- Adaptive - non-deterministic , per-hop based routing
- Delay model - no flit loss
- Distributed control - Each routing decisions locally.
- Relative addressing scheme – identical switch duplication, no memory unit

Each packet width is 128 bits out of which the first 96 bits are the data payload and the last 32 bits are the header which contains the destination address. The command and address sequence are stored in the payload as data.

3.1. Setup

There are 17 layers stacked in the model. The ground layer has an array of 4x4 microcontrollers and each of the next 16 layers have an array 4x4 flash memory banks. The microcontrollers inject request packets in every cycle to the network in 5 levels of rates; 10%, 30%, 50%, 70% and 90% of injection rates in uniform random traffic pattern (URT). 100% injection rate is when a packet is injected in every cycle. A packet hops from switch to switch in the network until it reaches the destination address specified in its header. In this case, the destination address is one of the memory banks in the top 16 layers. The addressed memory bank receives the packet and sends back an acknowledgement packet to the microcontroller. Upon the reception of acknowledgement packet, measurements of latency and throughput are taken. This process is done for 3300 cycles. Samples are taken for measurements between cycles 1500 – 3000 which are considered as steady state samples.

3.2. Results

3.2.1. Average Raw Latency

The average raw latency is the actual roundtrip time taken by the network to deliver a request packet to the memory bank and carry back the acknowledgement from the memory to the sender microcontroller. The microcontroller prepares a packet of 128 bit width with request. This packet is sent to a memory bank with destination address. Based on the destination address the packet adjusts itself to take the path that takes minimum trip distance.
But in fact, due to traffic congestion of the network, the packet follows a path which is slightly different from the ideal path. This is the actual raw latency. Figure 5 shows the average raw latency in terms of hops increases with increasing injection rate. This trend is expected because of the level of traffic congestion. When the injection rate is only 10%, on average, the raw latency is around 21.493 hops. But with increasing injection rate, to 50% then to 90%, the number of packets in the network increases and hence the network becomes more congested. In this case the latency increases to 26.309 hops for 50% and to 33.92 to 90% injection rates which is summarized in Table 1. For each hop, two cycles are needed. If the system is run in 48nm technology with clock frequency of 500 MHz, then the packet latency is 43 ns, 53 ns and 69 ns for 10%, 50% and 90% injection rates respectively. These network latencies are insignificant compared to the flash erase time of 200 us range. Thus, the flash erase time will continue to be the dominant factor in the overall latency in 3D models.

<table>
<thead>
<tr>
<th>Injection Rate</th>
<th>10%</th>
<th>30%</th>
<th>50%</th>
<th>70%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput per microcontroller (Packets/Cycle)</td>
<td>0.099</td>
<td>0.3</td>
<td>0.496</td>
<td>0.679</td>
<td>0.751</td>
</tr>
<tr>
<td>Normalized Latency (Actual Latency/Ideal Latency)</td>
<td>1.060</td>
<td>1.105</td>
<td>1.169</td>
<td>1.322</td>
<td>1.572</td>
</tr>
</tbody>
</table>

3.2.2. Average Normalized Latency

The actual raw latency tells us only the delay in the network. In order to measure the efficiency of the network, we need to compare the deviation of the actual latency from the ideal case. The normalized latency is the ratio of the actual to the ideal latency. The most efficient network has actual latency which the same as the ideal one. Hence, the normalized latency is 1. With increasing congestion level, the efficiency of the network decreases and has normalized latency > 1. Figure 6 shows the average normalized latency for increasing injection rate. The normalized latency at 10% injection rate shows efficiency close to the ideal case. With increasing injection rate, the congestion level increases. When 90%, the minimum normalized latency reaches around 1.6. Table 1, shows the average normalized latency measured for the sample cycles. The worst case is 1.572, when the injection rate is 90%. This latency is relatively acceptable, given that there are 16 microcontrollers working in parallel with 90% load.

![Figure 5: Average raw latency in hops per packet](image_url)
3.2.3. Average Throughput

The average throughput for each microcontroller is given in Table 1 for the different injection rate. The ideal throughput is equal to the number of packet released at the injection rate. The actual one is the measured output for each microcontroller node per cycle as described in [7]. For the 0.1 injection rate, the ideal throughput is 0.1 packets per cycle but Figure 7 shows that the actual is 0.099 which is < 1% margin. If we run the model with 500 MHz clock, then this throughput is equivalent to 0.099 * 500 MHz = 49.5Mpackets/sec. One packet is 128 bits width with 96 Data payload and 32 bit headers. Thus in terms of bandwidth, 49.5 Mpacket/sec * 96 bits = 4.752 Gbps. The throughput starts to slope down after 0.7 injection rate. The worst case scenario is when the injection rate is 0.9, which gives 0.751 packets per cycle or in terms of bandwidth, it is 36.048 Gbps. This is 16.5% less than the ideal and happens due to the relatively higher level of traffic congestion. Nevertheless, this bandwidth still shows that the network can effectively handle higher data rates.

![Average Normalized Latency](image)

Figure 6: Average Normalized Latency

3.2.3. Area

In 3D integration, a number of layers are stacked vertically. The total area of a 3D model is the foot-print of the largest layer. In the ELITE model, except the first layer with array of microcontrollers, all the top layers are stacked with array of flash memory banks as shown in Figure 8. Compared to the flash banks, the microcontroller area is tiny. Thus we take the flash bank area as the foot-print of the model. The arrangement in Figure 8 shows that for each NoC switch a flash bank with a total area of 25% of the largest footprint available. For example, the 8 Gbit QFL seen in Figure 1 is 150 mm². The NoC switch simulated in 180 nm technology is around 0.5 mm². If we extrapolate this area into 90 nm or better technology, the area of the NoC switch diminishes and becomes insignificant compared to the flash bank. Also, the TSV bundle contains 256 (128 up and 128 down) with diameter of 10 um and pitch of 20 um. The total area of TSV bundle is 0.11 mm² is yet in um² range and is tiny compared to the total flash bank area. Another factor is the switch to switch and switch to flash bank resource interconnects wires, which is around 10-20% of the total area [5]. Thus we can conclude that the area of a 3D flash memory is dominantly determined by the total area of flash banks placed in one die.
4. Conclusion

The total area of the 3D model is determined by the arrangement and size of the largest flash bank footprint. The data-write latency will continue to be significantly dominated by the time taken to erase the memory cell arrays. The network delay, due to planar wires, TSV and switch circuitry combined, is as such not significant, if one uses NoC as a means of communication between the microcontrollers and the memory blocks. Moreover, a high level scalability and parallelism can be gained by implementing NoC in 3D-structures. The simulation of the ELITE model shows that conceptually, it is possible to produce multi-layer, multi-processor 3D SSD with qualitative and quantitative performance excellence.

5. References