A Design Space Exploration of Binary-Tree-Decoder For Multi-Block- External-Memory-Controller in Multi-µc Embedded Soc Design

Uday Arun¹, A. K. Sinha²

¹, ²Bharath Institute of Science & Technology, Chennai, Tamil Nadu, India

Abstract—In any Micro-Controller Embedded System the whole memory is accessed by a single Micro-Controller. It uses only a fraction of memory and rest of the memory is wasted. In addition to this the only one Micro-Controller is executing all the given set of instructions or program. In our work we have designed a Multi-Block external memory & Binary-Tree-Decoder for a Multi-Micro-Controller Embedded System so that the complete memory is divided into more than one block (in our case it is 4) and it can be accessed independently by more than one Micro-Controller. This can be done in two ways one is static memory mapping mechanism and dynamic memory mechanism. In static memory mapping, the Micro-Controller is able to access only a particular block of memory to which it is mapped. While in the Dynamic memory mapping, any Micro-Controller can access any block of memory. Also, the different part program is executed by different Micro-Controller parallelly, which results in to speed up the execution speed of the Multi-Microcontroller system. Current embedded applications are migrating from single processor-based systems to intensive data communication requiring multi-processing systems to fulfill the real time application demands. The performance demanded by these applications requires the use of multiprocessor architecture. For these types of multiprocessor systems there is a need for developing such memory mapping mechanism which can support high speed. For selected memory mapping mechanism what should be the decoding mechanism and the controller design that gives low power consumption, high-speed, low- area system. Our algorithm of Binary-Tree-Decoder improves the MMSOPC embedded system design. However the designing of Binary-Tree-Decoder algorithm (for 256K memory) has not been designed by any of researcher which is presented in this work.

Keywords—SOC, MPSOC, MMSOC, MPSOPC

I. INTRODUCTION

Current embedded applications are migrating from single processor-based systems to intensive data communication requiring multi-processing systems to fulfill the real time application demands. The performance demanded by these applications requires the use of multiprocessor architecture. For these type of multiprocessor systems there is a need for developing such memory mapping mechanism which can support high speed. For selected memory mapping mechanism what should be the decoding mechanism and the controller design that gives low power consumption, high-speed, low- area system. The designing of an on chip micro networks meeting the challenges of providing correct functionality and reliable operation of interacting system-on-chip components [1]. The dynamic partitioning of processing and memory resources in embedded MPSoC Architectures [3]. The idea of dynamic partitioning of memory resources is been used in designing the architecture of multi-block memory. The MPSoC design challenges, one of the key MPSoC architectural point while designing and programming is the memory access methods [20].

To access the multi-block memory we have used dynamic memory mapping technique and designed a controller for this external memory that is used in 8-bit Multi-MicroController-System-On-Chip[20][3].

In a multi-microcontroller SOCs there can be a possibility of two kind of memory mapping.
- Static Memory mapping
- Dynamic memory mapping

A. STATIC MEMORY MAPPING:

In static memory mapping, the micro-controller is able to access only a particular block of memory to which it is mapped.

<table>
<thead>
<tr>
<th>Microcontroller No.</th>
<th>Memory Block address</th>
<th>Memory Block No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC01</td>
<td>0000-1FFF</td>
<td>BLOCK0</td>
</tr>
<tr>
<td>MC02</td>
<td>2000-2FFF</td>
<td>BLOCK1</td>
</tr>
<tr>
<td>MC03</td>
<td>3000-3FFF</td>
<td>BLOCK2</td>
</tr>
<tr>
<td>MC04</td>
<td>4000-4FFF</td>
<td>BLOCK3</td>
</tr>
</tbody>
</table>
B. DYNAMIC MEMORY MAPPING:

While in the Dynamic memory mapping any micro-controller can access any block of external memory, also the different program can be executed by different micro-controller parallely, which results in to high throughput & the performance of the multi-microcontroller system improves[19]. As Dynamic memory mapping gives more processing speed so we follow this technique[22]. But, this technique is very complex and requires extra hardware to implement this technique. An address decoding technique to generate memory address need to enhanced so that no extra burden on hardware.

<table>
<thead>
<tr>
<th>Microcontroller No.</th>
<th>Memory block No.</th>
<th>Control signal</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC01</td>
<td>Block0,Block1,Block2,Block3</td>
<td>RD=0 &amp; WR=1</td>
<td>READ</td>
</tr>
<tr>
<td>MC02</td>
<td>Block0,Block1,Block2,Block3</td>
<td>RD=0 &amp; WR=1</td>
<td>READ</td>
</tr>
<tr>
<td>MC03</td>
<td>Block0,Block1,Block2,Block3</td>
<td>RD=0 &amp; WR=1</td>
<td>READ</td>
</tr>
<tr>
<td>MC04</td>
<td>Block0,Block1,Block2,Block3</td>
<td>RD=0 &amp; WR=1</td>
<td>READ</td>
</tr>
</tbody>
</table>

In the Dynamic memory mapping, any micro-controller can access any block of memory.

---

A Design Space Exploration of Binary-Tree-Decoder For Multi-Block-External-Memory-Controller...
II. MEMORIES IN MULTI-MICROCONTROLLER ENVIRONMENT

There are different memory scheme available for SOCs having multiple Microcontrollers. Three of them are listed below[27].

- UMA (Uniform Memory Access)
- NUMA (Non-Uniform Memory Access)
- COMA (Cache Only Memory Access)
- CC-NUMA (Cache Coherency Non Uniform Memory Access)
- CC-COMA (Cache Coherency Cache Only Memory Access)

A. UMA (Uniform Memory Access) Model:

Physical memory is uniformly shared by all the processors. All the processors have equal memory access time to all memory words, so it is called uniform memory access.

B. NUMA (Non-Uniform Memory Access) Model:

NUMA multiprocessor is a shared-memory system where access time varies with the location of the memory word.
network. Besides distributed memories, globally shared memory can be added to a multiprocessor system. In this case, there are three memory-access patterns:

- (a) Local memory access (Fastest)
- (b) Global memory access.
- (c) Remote memory access (slowest)

In hierarchically structured multiprocessors, processors are divided into clusters which are itself an UMA or a NUMA multiprocessor. Clusters are connected to global shared-memory modules. All processors belonging to the same cluster are allowed to uniformly access the cluster shared-memory modules. All the clusters have equal access to the global memory.

C. COMA (Cache Only Memory Access) Model:

A multiprocessor using cache-only memory assumes the COMA model. This model is a special case of a NUMA machine, in which the distributed main memories are converted to caches. There is no memory hierarchy at each processor node. All the caches memory is accessible from a global address space. Remote cache access is assisted by the distributed cache directories (D).

D. Other Model:

Another variation of multiprocessors is a cache-coherent non-uniform memory access (CC-NUMA) model which is specified with distributed shared memory and cache directories. One more variation may be a cache-coherent COMA machine where all copies must be kept consistent.

---

**Fig. 5** COMA model of multiprocessor
A Design Space Exploration of Binary-Tree-Decoder For Multi-Block- External-Memory-Controller...

III. SOC MODEL AND DESIGN

This is the generic system architecture of the Multi- microcontroller system on chip. In our project the multi block external memory is basically the integrated form of the following components:
1. MAR (Memory Address Register)
2. MDR (Memory Data Register)
3. MMBAC(Multi Memory Block Arbitration Controller)
4. BINARY TREE DECODER
5. Memory Block comparator
6. Memory of 256 Kb

In the above shown architecture the four microcontroller are connected to the memory via different controllers and buffer (register like MAR, MDR) with the help of address bus , data bus and control bus. Buses are nothing but the set of parallel wires that connect two components. In this architecture the data buses are of 8Bits, address buses are of 16Bits wide.

A. BINARY TREE DECODER
In the above implementation the binary tree decoder is the block which helps to implement the low power dynamic memory accesses by keeping in mind the low power and area.
A Design Space Exploration of Binary-Tree-Decoder For Multi-Block-External-Memory-Controller...

After Decoding Mechanism of Binary Tree Decoder, Binary Tree generation and memory address generation for all blocks of memory follows an algorithm which is as follows:

\[
\text{for } i \text{ in 0 to 255 loop}
\text{ if } \text{conv_integer(BinaryTreeDecoder_En)} = i \text{ then}
\]

\[
\text{---ForestTreeDecoder_Out := 16*i + conv_integer(y);} \quad \text{-------------Old Algo.}
\text{BinaryTreeDecoder_Out := 256*i + conv_integer(y); \quad \text{-------------New Algo.}}
\]

B .RTL Schematic of Decoding Mechanism of Binary Tree Decoder

![RTL Schematic of Binary Tree Decoder (BTD)]
A Design Space Exploration of Binary-Tree-Decoder For Multi-Block-External-Memory-Controller...

Fig. 8: RTL schematic of integrated block

Fig. 9: RTL schematic of MULTI-BLOCK-EXTERNAL-Memory-Controller IN Multi-µC SOC DESIGN

C. Results
The memory controller design using binary tree decoder is implemented on FPGA board and below table summarises the result.
C.1. Design summary of Binary tree decoder

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Forest Tree Decoder</th>
<th>Binary Tree Decoder</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Decoder</td>
<td>4096</td>
<td>256</td>
<td>16 times less than the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Forest Tree Decoder</td>
</tr>
<tr>
<td>Synthesis Time/Issues</td>
<td>Unable to finish</td>
<td>Able to synthesize</td>
<td></td>
</tr>
<tr>
<td></td>
<td>synthesis, tool</td>
<td>faster</td>
<td></td>
</tr>
<tr>
<td></td>
<td>crashes due to</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>requirement of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>large virtual</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>memory space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>More number of</td>
<td>Less number of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>gates</td>
<td>gates</td>
<td></td>
</tr>
</tbody>
</table>
C.3. Power Analysis of Binary Tree Decoder

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Performance metric (power)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
</tr>
<tr>
<td>1</td>
<td>Vccint</td>
</tr>
<tr>
<td>2</td>
<td>Vcco33</td>
</tr>
<tr>
<td></td>
<td>Total Power</td>
</tr>
</tbody>
</table>

C.4. Results from XPower summary report of the integrated design

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Total Current</th>
<th>Total Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vdd =3.3V, Vt =1.8V</td>
<td>17mA</td>
<td>33.6mW</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

Our result is tested, verified & prototyped on the following environment using following Hardware & software tools & technology:

- **Target Device**: xc2s600e-6fg456
- **On Board frequency**: 50 MHz
- **Test frequency**: 12MHz
- **Supply Voltage**: 3.3 volts
- **Worst Voltage**: 1.4 to 1.6 volts
- **Temperature**: -40°C to 80°C
- **Speed Grade**: -6
- **FPGA Compiler**: Xilinx ISE 8.2i
- **Simulator used**: ModelSimXE-ii 5.7g

The initial implementation was based on forest tree decoder. The major enhancement done in this work is to replace Forest Tree Decoder with Area, power efficient implementation of Binary Tree Decoder. The decoding process is one of the most time consuming process, this decoding mechanism in turn makes the whole MPSOC system faster and efficient with help of multi block memory architecture. With reference to the 2d VLSI complexity model we have found that as x-axis (data lines) width is kept fixed (8 bit) and y-axis (row address of memory) height is reduced in case of binary tree decoder in comparison to forest tree decoder. So the over all area of 2D model is reduced which in turn increases the accessing speed of the multi-processor system. As the y-axis height is reduced this implies that searching time of particular memory word (8 bit here) is reduced hence accessing speed is increased. In another aspect, as the area is lesser and the power consumed is also lesser.

REFERENCES

Journals:
A Design Space Exploration of Binary-Tree-Decoder For Multi-Block- External-Memory-Controller...