EE201A Presentation

Memory Addressing Organization for Stream-Based Reconfigurable Computing

Team member:
- Chun-Ching Tsan: Smart Address Generator - a Review
- Yung-Szu Tu: TI DSP Architecture and Data address

Outline – Smart Address Generator

3. Address Generation Unit (AGU) (1991)
4. GAG (generic address generator) (1990)
5. GAG of MoM-2 (1991)


CPU-Memory Model: a von Neumann machine
- computational processor (CP)
- memory access processor (MAP)

A Simple Example for Image Processing

- The needed address patterns are generated by a dedicated counters or circuit transformations applied to a counter output.
Logic Synthesis for Semi-Random Address Sequences

Address Generation Unit (AGU) (1991)
- an application specific address generation unit for video signal processor (VSP), a specified DSP
- implementing a 2-level address generation with window based memory access, without full slider method.
- 3 AGUs running in parallel calculate the address for external image memory
- Providing 17 addressing modes:
  - a 2-D raster scan mode
  - a block scan mode for spatial filtering
  - 8 variants of a neighborhood search mode
  - a 2-D indirect access mode for external image memory
  - a FFT mode and an affine transformation mode

DSP Architecture

GAG (generic address generator) (1990)
- MoM-1 (Map-oriented Machine 1)
  - an image processing machine with 2-D memory organization
  - implement a pattern matching approach
  - avoiding address calculation overhead and fully parallelized pattern matching by
    a a dynamically reconfigurable PLA (DPLA)
  - address generator: move control unit (MCU)
    - an application specific generic address generator
    - configured before execution time
    - needs no memory cycles at run time

GAG (generic address generator) (1990)
The MoM xputer architecture
Mapping Application and Memory Communication
Texas Instruments
TMS320C54x
DSP
Architecture and Data Addressing

Class presentation of EE201A
May 16, 2003

Agenda
• Architecture
• Block diagram
• Immediate addressing
• Absolute addressing
• Accumulator addressing
• Direct addressing
• Memory-mapped register addressing
• Stack addressing
• Indirect addressing
• Reference

Architecture

• Advanced Harvard architecture
  – Separate data and program memory allows a high degree of
  parallelism
• CPU can read and write to a single block in the same cycle

Block Diagram

• Memory Access
  – 4 internal bus pairs
  – CD for data read
  – E for data write
  – P for program
• Others
  – 2 40-bit Accum.
  – 40-bit Barrel shifter
  – 40-bit ALU
  – 17x17 multiplier and 40b dedicated
  adder perform a non pipelined
  single-cycle MAC

Immediate and Accumulator Addressing

• The instruction syntax contains the specific value of the
  operand
  – LD #80h, A
• Immediate values can be 3, 5, 8, 9, or 16 bits in length

Figure 6-1. RPT Instruction With Short Immediate Addressing

<table>
<thead>
<tr>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 6-2. DPT Instruction With 16-bit Immediate Addressing

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• Accumulator addressing
  Uses the accumulator as an address
  – READA Smem

Absolute addressing

• Addresses are always 16 bits long, addressing types
  depend on instructions
• Data-memory address (dmad) addressing uses a specific
  value to specify an address in data space
  – MVKD SAMPLE, *AR5
• Program-memory address (pmad) addressing uses a
  specific value to specify an address in data space
  – MVPD TABLE, *AR7
• Port address (PA) addressing uses a specific value to
  specify an external I/O port address
  – PORT FIFO, *AR5
• ‘(ik) addressing uses a specific value to specify an
  address in data space
  – Instructions with single data-memory operand
  – LD *(BUFFER), A
Direct addressing

- Uses the accumulator as an address
  - READA Smem
- With direct addressing, instructions contain the lower 7 bits of the data-memory address (DMA)
  - Combined with a base address, data-page pointer (DP) or stack pointer (SP) to form a 16-bit data-memory address
  - ADD SAMPLE, B
  - DR referenced
  - SP referenced

Memory-mapped register addressing

- Used to modify the memory-mapped registers without affecting the current data-page pointer (DP) or stack pointer (SP)
  - Overhead for writing to a register is minimal
  - Works for direct and indirect addressing
  - SCRATCH-PAD ram located on data page 0 can be modified
  - STM #x, DIRECT
  - STM #tbl, AR1

Stack addressing

- Used to automatically store the program counter during interrupts and subroutines
- Can be used to store additional items of context or to pass data values
- Uses a 16-bit memory-mapped register, the stack pointer (SP)
- PSHD X2

Indirect addressing

- 8 auxiliary registers (AR), and 2 auxiliary register arithmetic units (ARAU)
  - Circular address modifications (MOD=8,9,10,11 or 14) for convolution, correlation, FIR filters, etc.
    - Circular buffer is a sliding window containing the most recent data
    - Circular buffer size register (BK) specifies the size of the circular buffer
      - Circular buffer of size R must start on a N-bit boundary, where \( 2^N > R \)
      - BK=32
      - Index is the N LSBs of ARx
      - Index is incremented or decremented by step

Indirect addressing (cont’d)

- Circular address modifications (MOD=8,9,10,11 or 14) for convolution, correlation, FIR filters, etc.
  - Circular-buffer size register (BK) specifies the size of the circular buffer
    - Circular buffer of size R must start on a N-bit boundary, where \( 2^N > R \)
    - BK=32
    - Index is the N LSBs of ARx
    - Index is incremented or decremented by step
Indirect addressing (cont’d)

- Bit-Reversed Address Modifications (MOD=4 or 7)
  - Enhances execution speed and program memory for FFT algorithms that use a variety of radices
- Assume FFT size is $2^N$, then AR0 = $2^{N-1}$
  - An ARx points to the physical location of a data value

<table>
<thead>
<tr>
<th>AR0: CB</th>
<th>AR1: CB</th>
<th>(n-1)th Value</th>
<th>(n-2)th Value</th>
<th>(n-3)th Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110 000</td>
<td>0110 100</td>
<td>1000</td>
<td>1100</td>
<td>1110</td>
</tr>
<tr>
<td>0110 100</td>
<td>0110 000</td>
<td>0110 000</td>
<td>1110</td>
<td>1110</td>
</tr>
<tr>
<td>0110 000</td>
<td>0110 000</td>
<td>0110 000</td>
<td>1110</td>
<td>1110</td>
</tr>
</tbody>
</table>

References

- Texas Instruments TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals(SPRU131)
- Texas Instruments TMS320C54x DSP Reference Set, Volume 2: Mnemonic Instruction Set(SPRU72B)