ARM7TDMI Processor Implementation in Verilog

Sasank Chilamkurthy, Varun Bairaboina, Bhanu Vikas, Nithin Nethipudi

May 2014

Abstract

We have implemented pipelined version of ARM v7 processor.

1 Pipeline stages design

We have 6 pipeline stages:

1. Instruction Fetch (F)
2. Register Read (R)
3. Multiplier (M)
4. ALU
5. Memory (MEM)
6. Writeback (W)

1.1 Instruction Fetch (F)

We fetching instructions from instruction cache and determine type of instruction. There is not much complex logic here.

1.2 Register Read (R)

We read all poosible registers that can be needed by a instruction.Also, forwarding decisions are made here.

1.3 Multiplier (M)

Multiplier block does any possible multiplications required.

1.4 ALU

This stage has ALU, Barrel shifter and condition field checker. Flag outputs from ALU is forwarded to the same stage i.e. fed back to itself.

1.5 Memory (MEM)

Accesses memory for Load/store instructions.

1.6 Writeback

We write back to registers and PC depending on instruction.

2 Implementation details

We made different modules for each blocks mentioned above and instantiated them in top level controller module. We maintained a `will_this_be_executed` signal at each pipeline stage to facilitate easier flushing and bubble insertion. Code is loaded by editing `initial` block of instruction cache module. We checked design with the standard fibonacci sequence generator program.