Hyper Massively Parallel Processor (HMPP) Array and CMOS Imager

Dr. C. H. Ting
Offete Enterprises

September 23, 2006

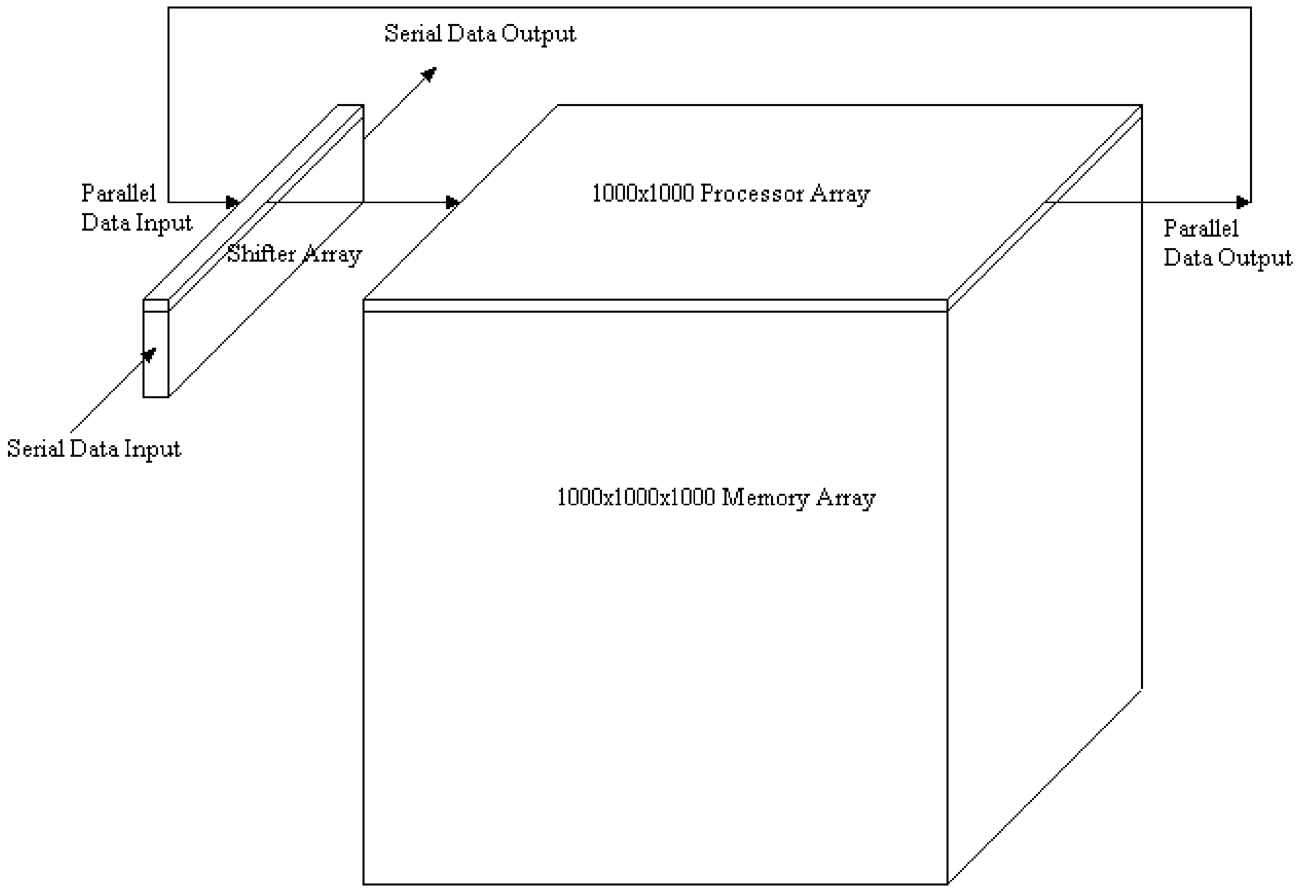
Summary

- HMPP (Hyper Massively Parallel Processors)
- Controlling HMPP
- CMOS Imager MT9M1111
- HMPP-Imaging System
- Imaging Finite State Machine
- Programming HMPP
- Demonstrations

Hyper Massively Parallel Processors

- One million processors in 1000x1000 array
- Single Instruction Multiple Datapath (SIMD) architecture
- At 300 MHz clock rate, the computation throughput is 300,000 GIPS

HMPP Architecture

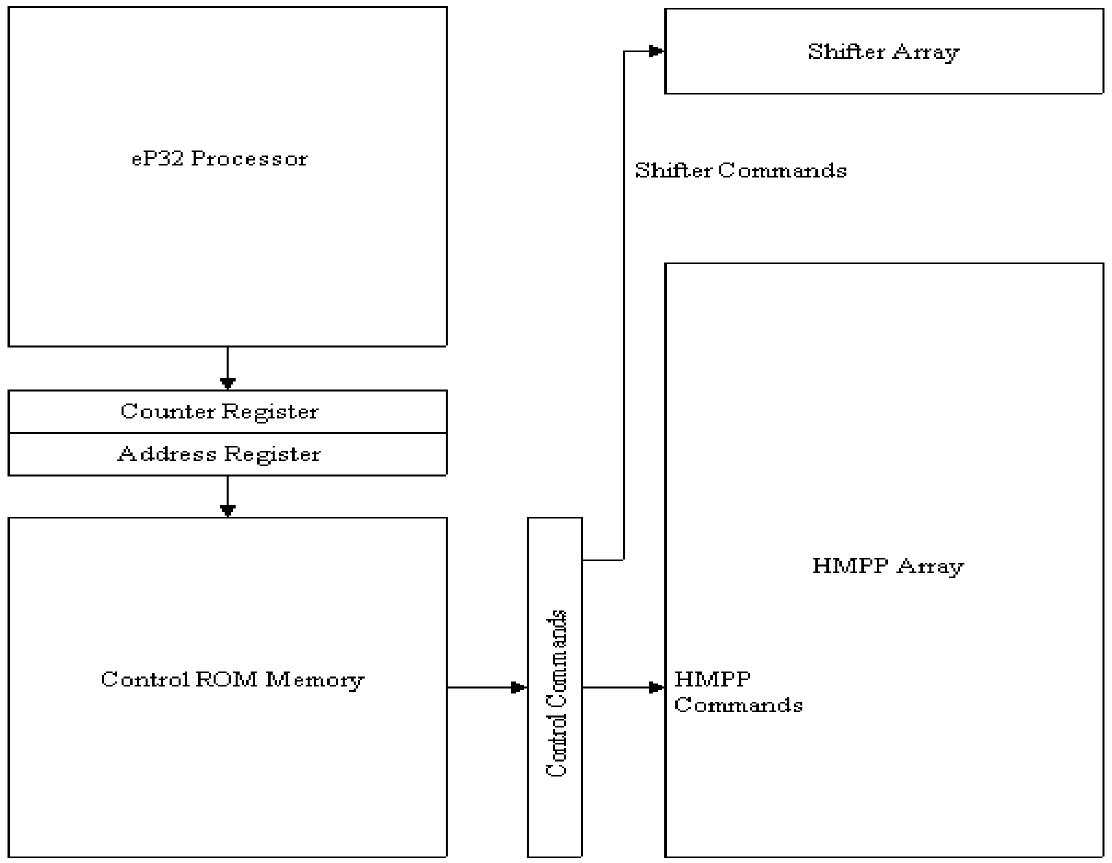


HMPP (Hyper Massively Parallel Processors)

Controlling HMPP

- ePP32 serves as master
- All HMPP instructions are stored in a Control_ROM memory
- Ease in Software Development
- A set of registers to control HMPP
  - ADDRESS_REG
  - COUNTER_REG
  - CMD_FRAMES
  - IO_CONTROL

Controlling HMPP



Controlling HMPP Array

CMOS Imager MT9M1111

- Produced by Micron
- 1.3 Mega Pixels
- Completely integrated camera system
- I2C Interface to computer
- Extremely complicated with 500 registers to control:
  - RGB sensor array
  - Camera control
  - Color pipe processor

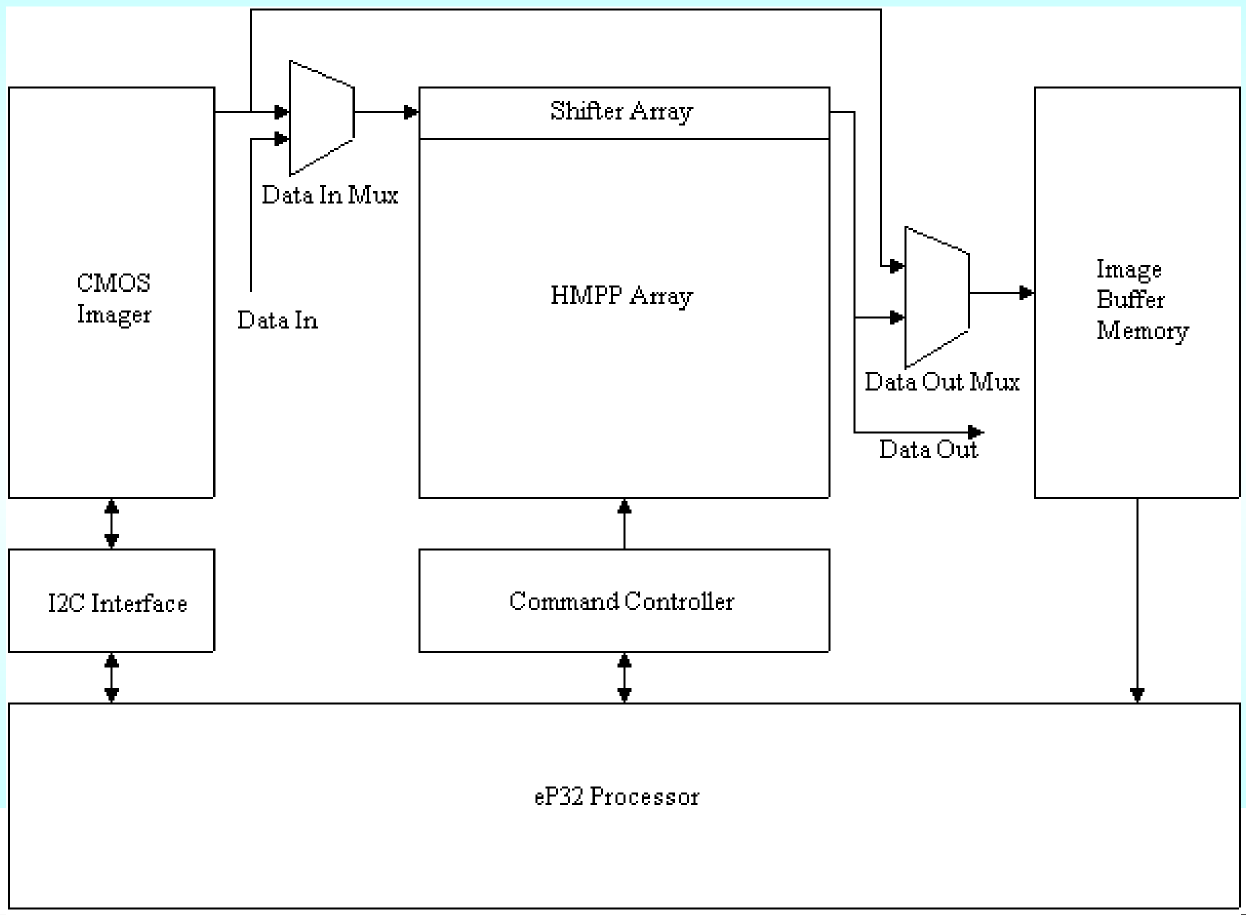
CMOS Imager MT9M1111

- Very easy to manage this complicated system in FORTH
- Two FORTH words:
  - @@	Read from a register
  - !!	Write to a register

HMPP-Imaging System

- ePP32, HMPP array and CMOS Imager are all driven by the same 16 MHz clock
- eP32 reads and writes registers in HMPP controller
- eP32 reads and writes registers in CMOS Imager
- CMOS Imager is free-running, sending out video images
- A Finite State Machine schedules HMPP processing on video images.

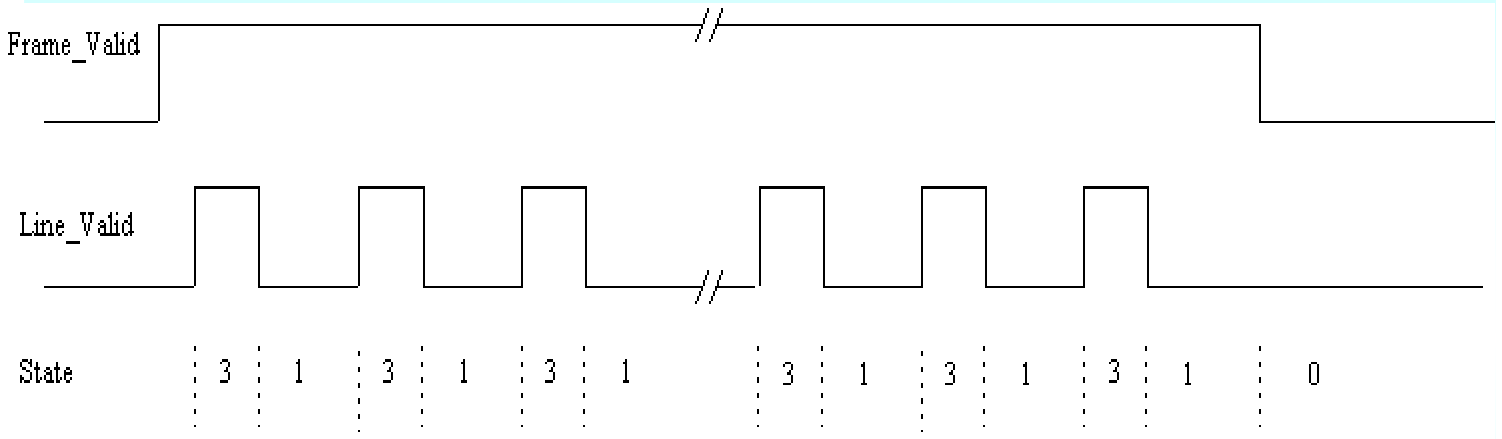
HMPP-Imaging System



Imaging Finite State Machine

- Running at 16 MHz, while synchronized to the 8 MHz pixel clock from CMOS Imager
- 3 States defined by FRAME_VALID and LINE_VALID signals from CMOS Imager
- Force HMPP to execute these routines:
  - SHIFT: to capture pixel data into Shifter Array in horizontal scan line
  - ROLL: to roll one line of data from Shifter Array into HMPP ARRAY in horizontal retrace
  - NEW_FRAME: to initiate a new integration in vertical retrace
  - INTEGRATE: to accumulate current image in vertical retrace

Imaging Finite State Machine



State	Description
0	Vertical Retrace, execute NEW_FRAME or INTEGRATE
1	Horizontal Retrace, execute ROLL
2	Invalid State
3	Horizontal Scan, execute SHIFT

State Machine to Acquire Images

Programming HMPP

- HMPP Assembler produces ROM image of HMPP instructions in a control_rom.mif file
- eP32 Metacompiler produces RAM image of FORTH system in a mem.mif file
- Quartus II synthesizes the complete FPGA chip design file
- eForth system allows interactive programming of HMPP array

Programming HMPP

- SHIFT: A single instruction at address 0 repeated 192 times
- ROLL: 33 instructions at address 3
- NEW_FRAME: 106 instructions at address 200H
- INTEGRATE: 121 instructions at address 300H

Demonstrations

- Quartus II design environment
- Synthesize ep32-HMPP system
- eForth system on eP32
- CMOS Imager operations
- HMPP image integration operations
- HMPP programming

Thank you very much!