PixelPerfect

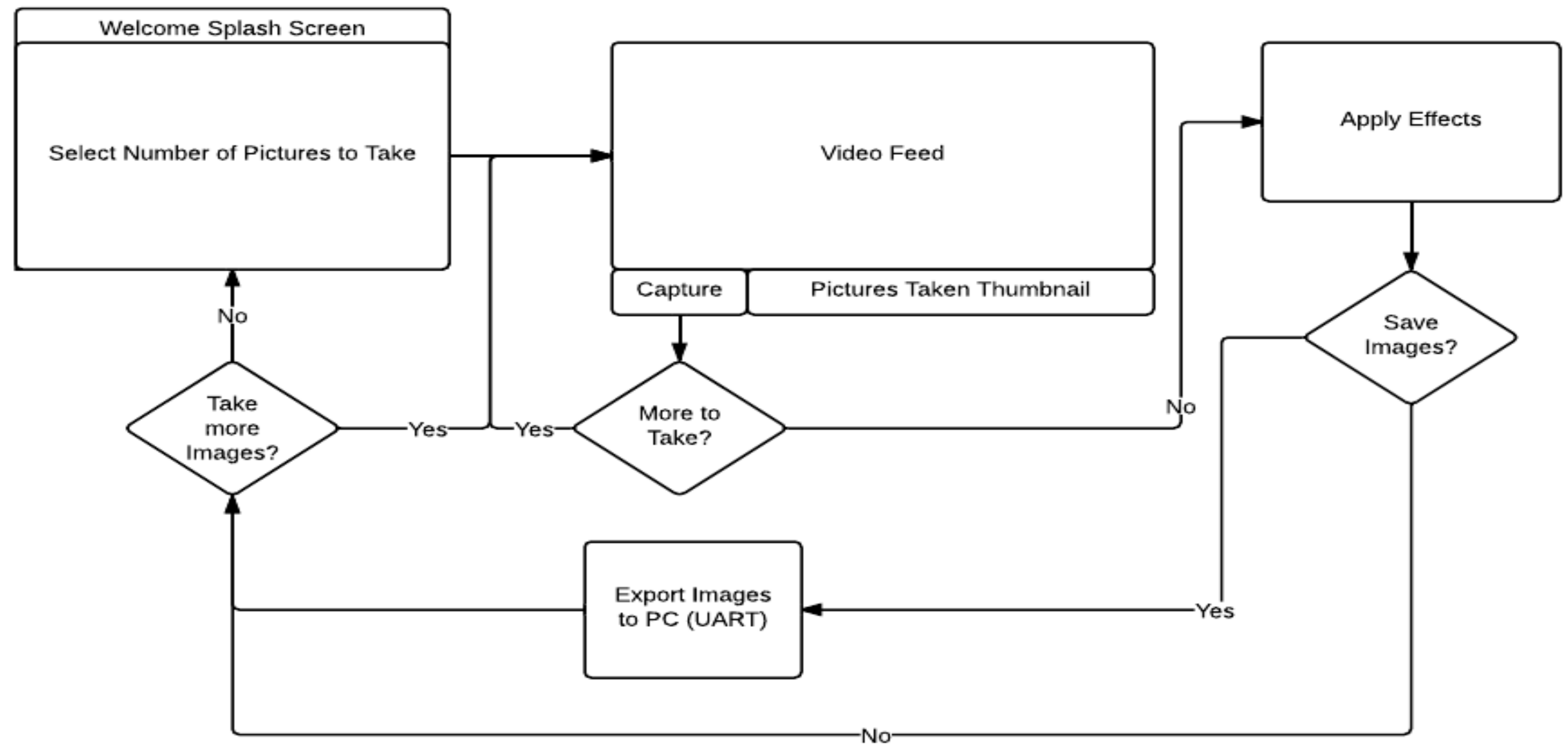
Kevin Avery, Scott Daw,
Mac Wibbels, Trevor Hill

Project Overview

- Photobooth
  - Take photos on FPGA
  - Apply optional effects
  - Send to computer
    - Print or post online

- Outline
  1. Application
  2. I/O
  3. Assembler
  4. Processor

Application



I/O

- RAM
- VGA
- UART
- Camera
- Interface Buttons

I/O - RAM

- "Cellular RAM"
  - Pseudo Static RAM Chip
  - Operates at 50MHz
  - 2^23 address * 16-bits/address = 16MB
  - Sophisticated interfacing
    - Burst reads/writes up to 128 addresses (1 physical RAM line)
- Used for full direct mapped screen frame buffer
  - 320 * 240 * 16-bits ~= 1.2MB
  - 2 pixels per address
- Also used for program stack and heap
- Need to arbitrate between VGA burst reading, processor read/writing, camera burst writing

0 - 127
128 - 255
255 - 383
384 - 511
...

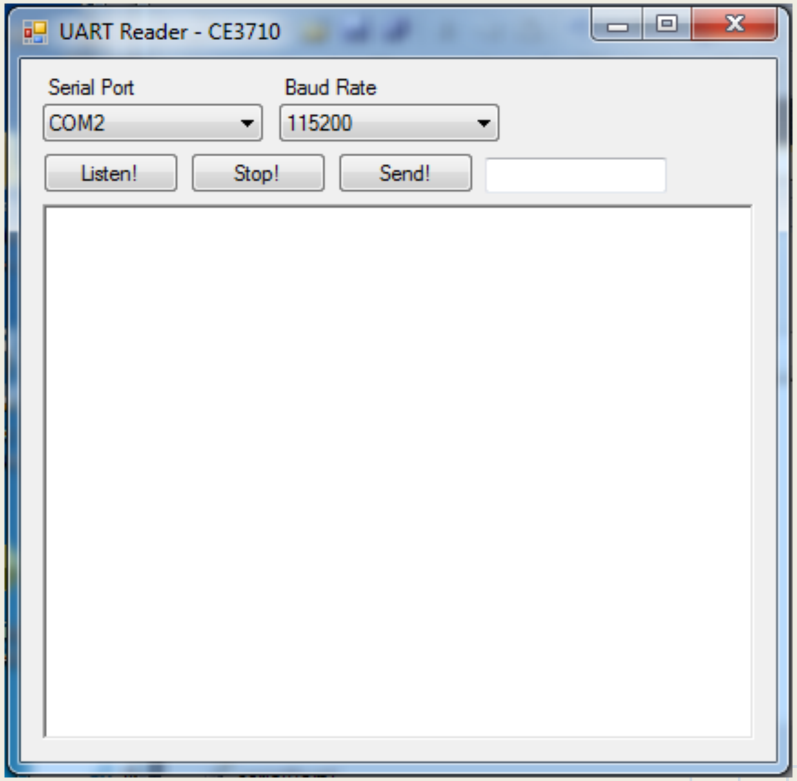
I/O - VGA

- Color
  - 8-bit: 3 Red, 3 Green, 2 Blue
- Resolution
  - Originally planned to use 800x600 at 40MHz
  - Caused timing related fuzziness/artifacts
  - Switched to 640x480 at 25MHz - life is good
- Buffered video pipeline
  - 1024 pixel dual-ported BRAM buffer
  - Burst read from RAM at 50MHz
  - Drawn to screen at a 25MHz

0 - 127	**128 - 255**	**256 - 383**	**384 - 511**

I/O - UART

- Transmit pictures from FPGA to Computer
  - Photobooth-like printing
  - User controlled camera options
  - Build bitmap image from UART data

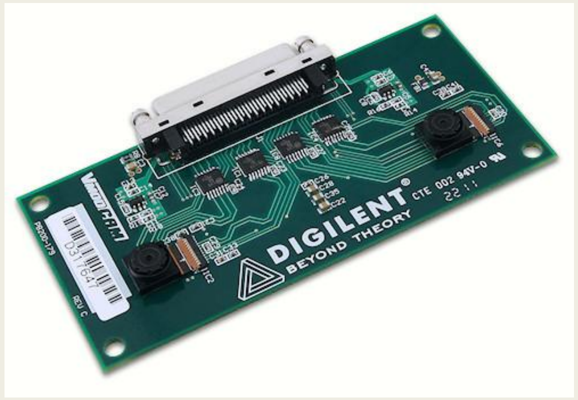

UART Reader - CE3710

Serial Port: COM2

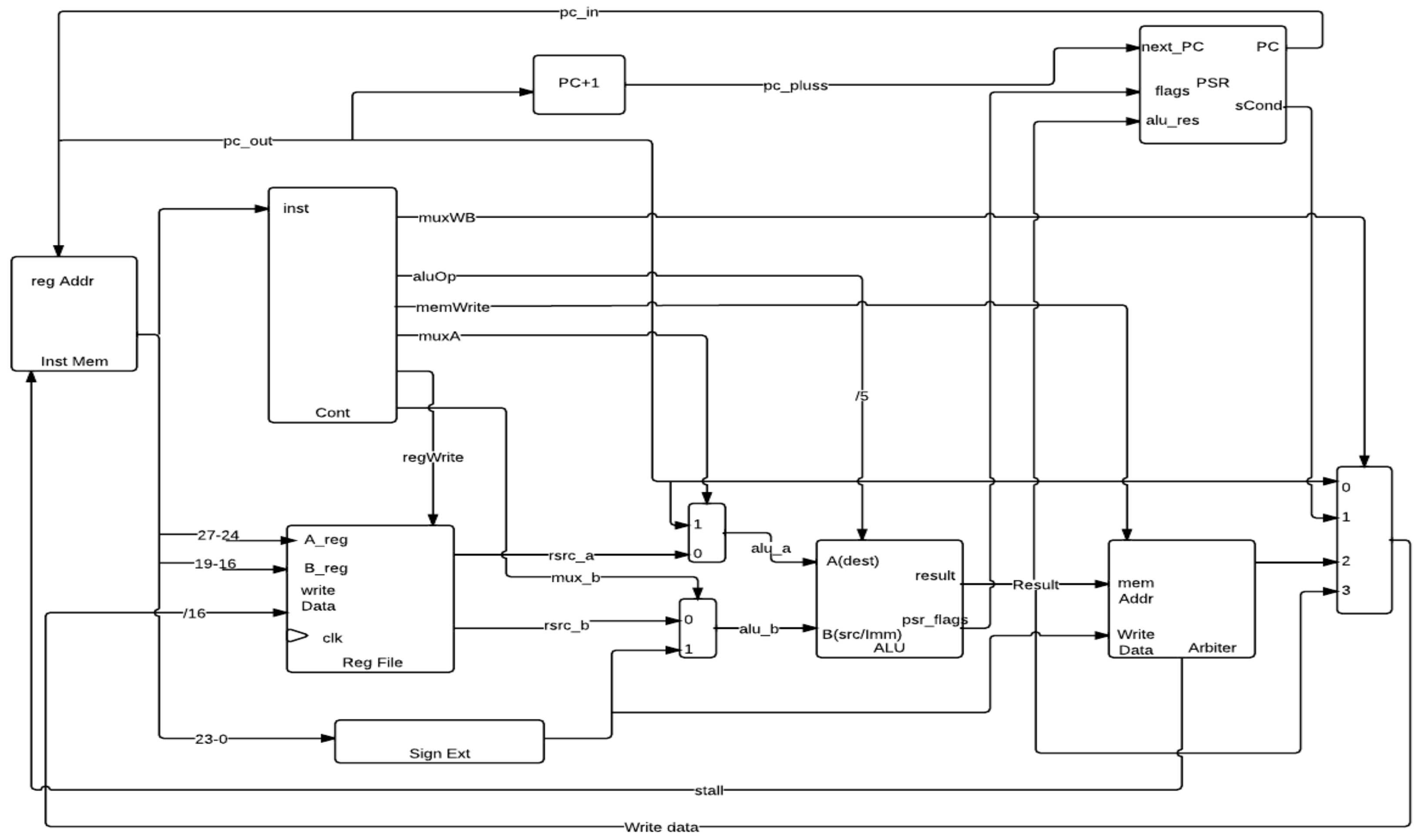
Baud Rate: 115200

I/O - Camera

- Hard
- Configuration
  - Write specific values into predefined registers on camera chip
  - Variable clock configurations, resolutions, color formats, correction options, more...
  - I2C two-wire interface
- Buffering
  - Sync data from camera pixel clock with RAM clock
  - Burst write when VGA not burst reading



Processor


pc_in
PC+1
pc_pluss
next_PC
PC
flags
PSR
sCond
alu_res
pc_out
inst
muxWB
reg Addr
aluOp
memWrite
muxA
Inst Mem
Cont
/5
regWrite
0
1
2
3
27-24
A_reg
19-16
B_reg
write Data
/16
clk
Reg File
rsrc_a
mux_b
rsrc_b
alu_a
alu_b
A(dest)
result
B(src/Imm)
psr_flags
ALU
Result
mem Addr
Write Data
Arbiter
23-0
Sign Ext
stall
Write data

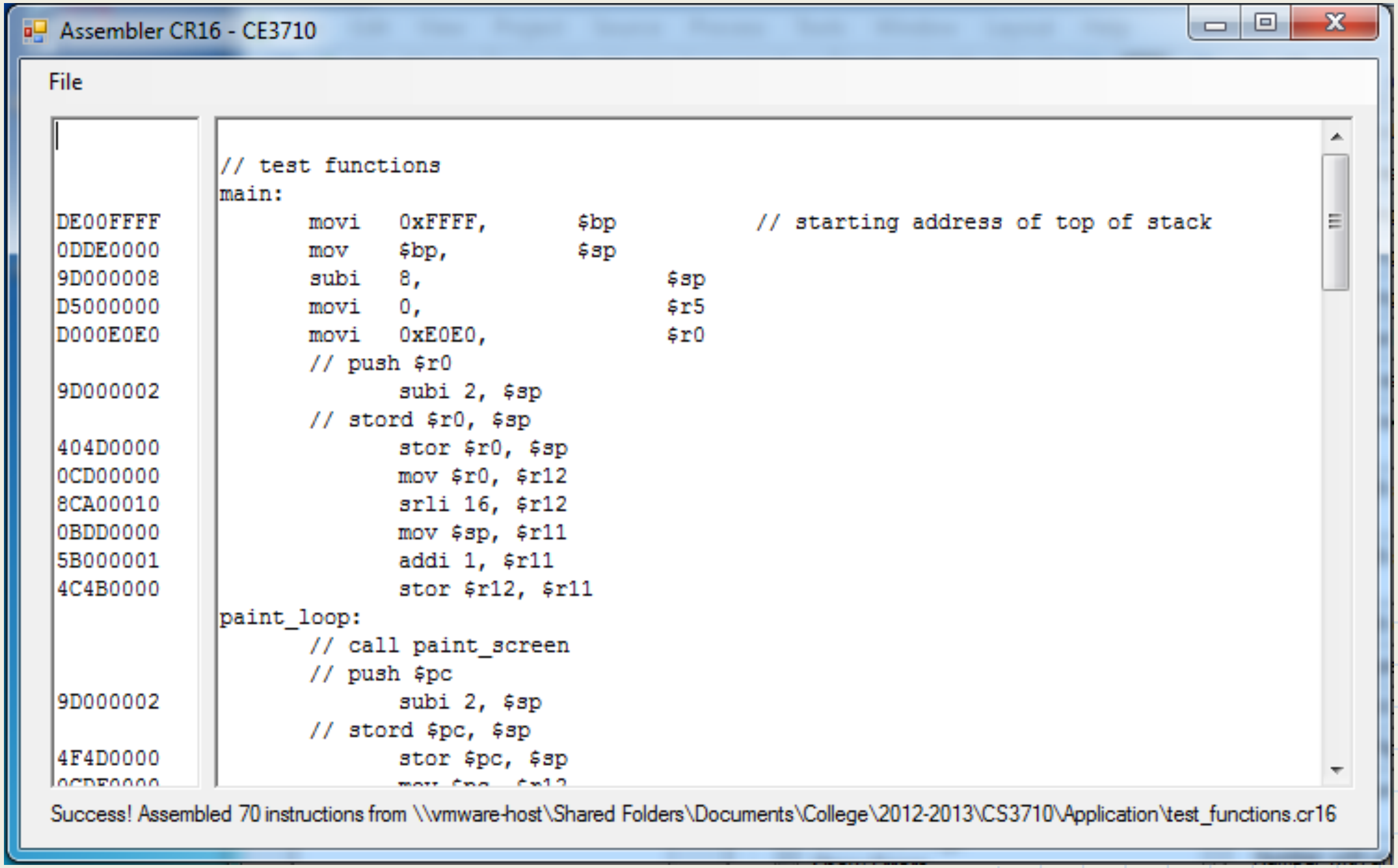
Processor

- Single Cycle Design
  - No state machine
  - Can operate at 75MHz

- Extended datapath to 32-bit
  - Using 16 bits allows for 64K addresses
  - Single image frame is 75K addresses
  - I-type instructions now have 24 bit immediates allowing for addressing to all 8Meg addresses of RAM
  - Extra upper bits are used for memory mapped I/O as needed

- Uses stalls
  - Multiple applications accessing memory
  - Fixed VGA checkering effects

Assembler



```
                // test functions
                main:
DE00FFFF                movi   0xFFFF,      $bp           // starting address of top of stack
0DDE0000                mov    $bp,         $sp
9D000008                subi   8,                  $sp
D5000000                movi   0,                  $r5
D000E0E0                movi   0xE0E0,             $r0
                        // push $r0
9D000002                       subi 2, $sp
                        // stord $r0, $sp
404D0000                       stor $r0, $sp
0CD00000                       mov $r0, $r12
8CA00010                       srli 16, $r12
0BDD0000                       mov $sp, $r11
5B000001                       addi 1, $r11
4C4B0000                       stor $r12, $r11
                paint_loop:
                        // call paint_screen
                        // push $pc
9D000002                       subi 2, $sp
                        // stord $pc, $sp
4F4D0000                       stor $pc, $sp
```

Success! Assembled 70 instructions from \\vmware-host\Shared Folders\Documents\College\2012-2013\CS3710\Application\test_functions.cr16

Assembler

- 5 Phases
  - Load in .cr16 assembly file
  - Decompose pseudo instructions
  - Map labels to offsets
  - Encode mnemonic instructions to hex
  - Write hex instructions to .dat or .coe

- Customized ISA
  - Added several pseudo instructions for function calling/stack manipulation
  - Simplified shift commands
  - Removed unnecessary ADDC/SUBC commands to free up OP-codes
  - Expanded to support 32 bit instructions

Questions?