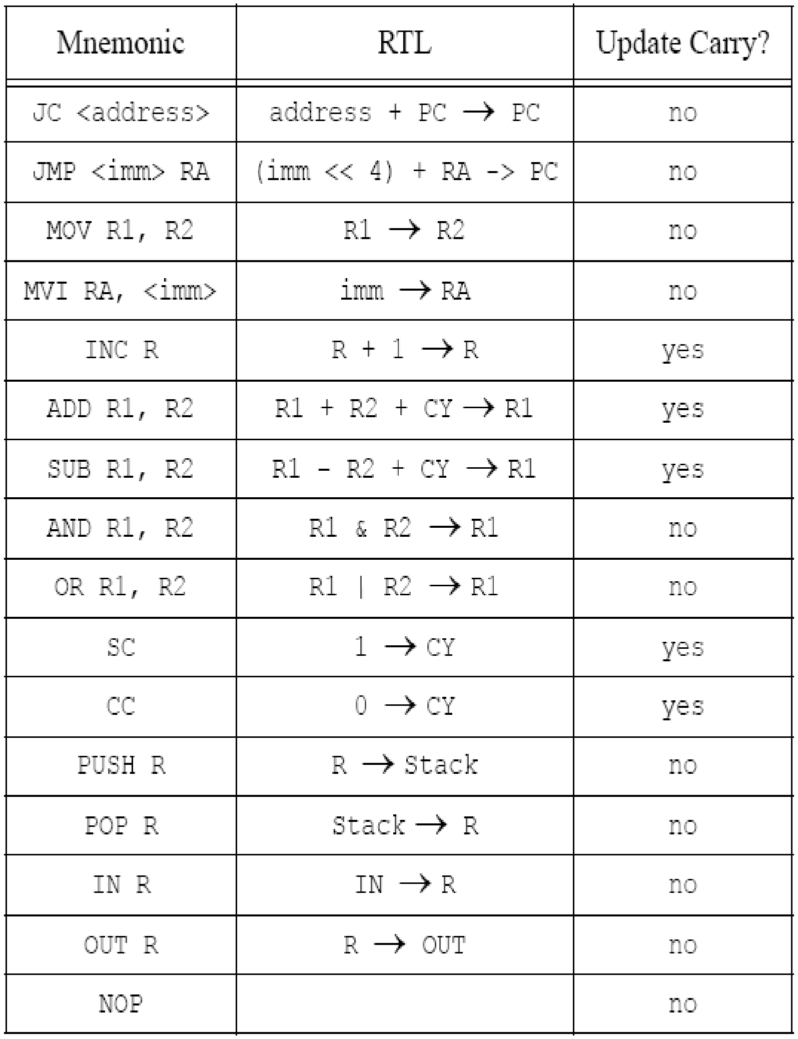
Basic **U**ndergraduate **R**isc **P**rocessor – BURP

Introduction & Design

Instruction Set



Mnemonic	RTL	Update Carry?
JC <address>	address + PC → PC	no
JMP <imm> RA	(imm << 4) + RA -> PC	no
MOV R1, R2	R1 → R2	no
MVI RA, <imm>	imm → RA	no
INC R	R + 1 → R	yes
ADD R1, R2	R1 + R2 + CY → R1	yes
SUB R1, R2	R1 - R2 + CY → R1	yes
AND R1, R2	R1 & R2 → R1	no
OR R1, R2	R1 | R2 → R1	no
SC	1 → CY	yes
CC	0 → CY	yes
PUSH R	R → Stack	no
POP R	Stack → R	no
IN R	IN → R	no
OUT R	R → OUT	no
NOP		no

Block Diagram (rough)

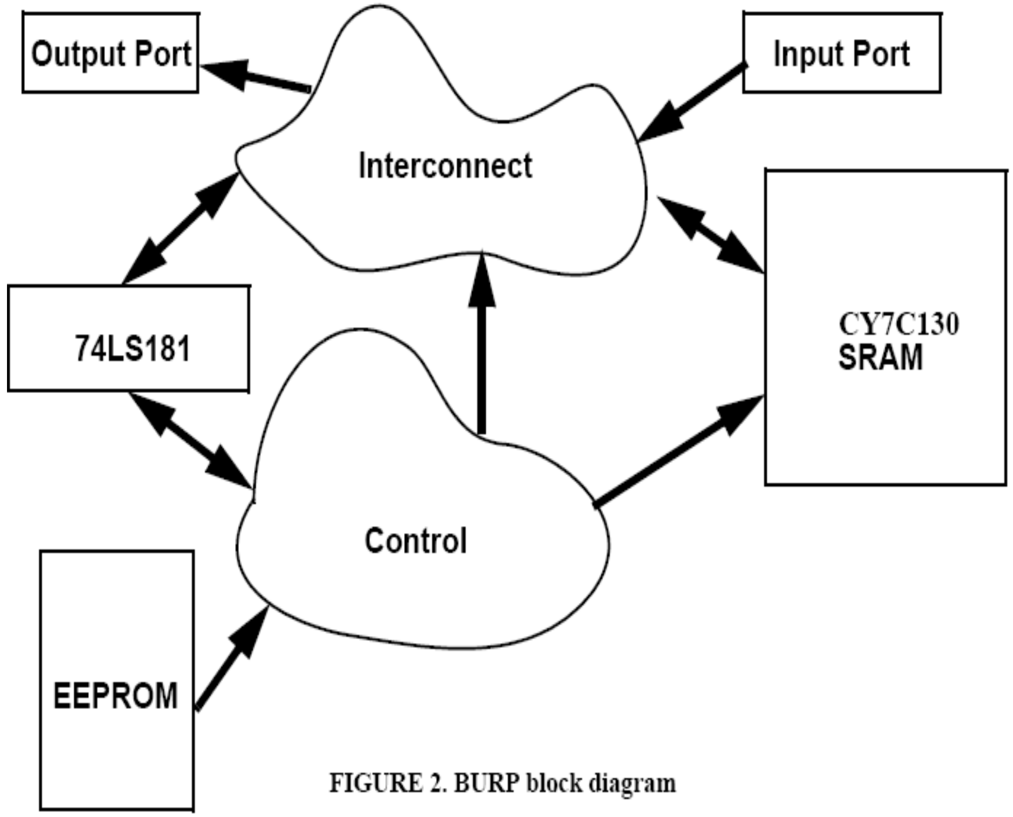


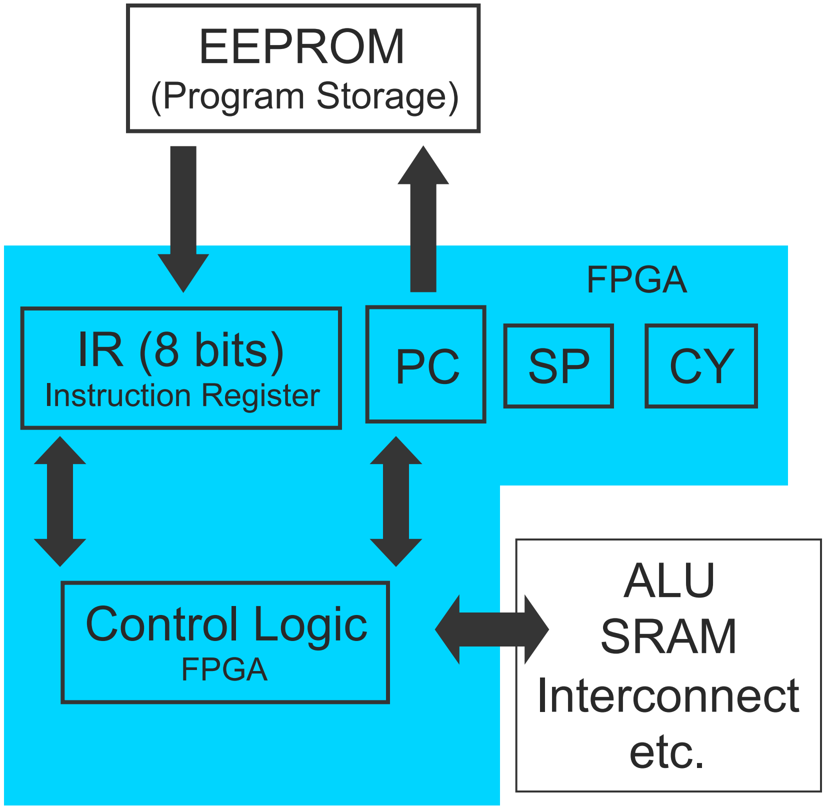
FIGURE 2. BURP block diagram

Features

- Data Path
  - 4-bit data, 1-bus or 2-bus design
  - ALU for logical & arithmetic operation
- Registers & Stack (SRAM)
  - 4 data registers
  - 8 stack locations
- Program Storage (EEPROM)
  - 8-bit address (256 memory locations)
  - 8-bit instruction words
- I/O Ports
  - Input: DIP switches
  - Output: register & 4 LEDs

Control Unit

- Registers
  - Instruction Register (IR)
  - Program Counter (PC)
  - Carry Register (CY)
  - Stack Pointer (SP)
- Control Logic
  - Signals to other chips (ALU, SRAM, I/O etc.)


EEPROM (Program Storage)

FPGA

IR (8 bits) Instruction Register

PC

SP

CY

Control Logic FPGA

ALU SRAM Interconnect etc.

Program Storage (EEPROM)

- 8-bit address from PC
- 8-bit instruction words
  - 4-bit opcode: 1 of 16 instructions
  - 4-bit data / operands:
    - Immediate value (4 bits)
    - Register fields: R1, R2 (2 x 2 bits)
      - Source & destination registers

Program Counter (PC)

- Address range: 2x4 = 8 bits
  - two 4-bit registers: PC = {PC$_{high}$, PC$_{low}$}
  - PC$_{high}$, PC$_{low}$ individually accessible
- Implementation
  - T
    i
  - T t)

You can implement it in FPGA!

Stack & Stack Pointer

- Stack
  - ≥ 8 stack locations
  - Status detection: full / empty
- Stack Pointer (SP): ≥ 3 bits
  - Up-/down-counter (e.g. ‘193)
  - Register (use ALU to inc./dec.)
  - FPGA (address output to MUX/SRAM)

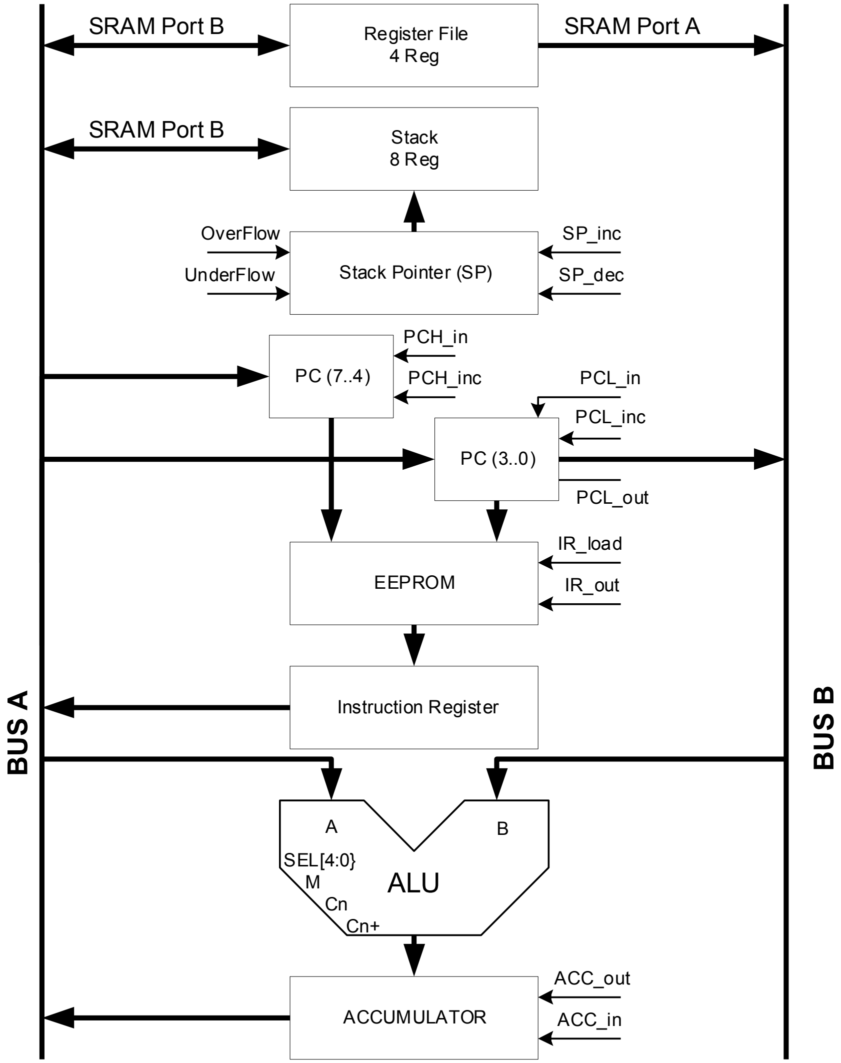
SRAM (Register File & Stack)

- Registers & stack are SEPARATE, unique memory locations
- Two Ports
  - 1-bus design: read-port, write-port
  - 2-bus design: read/write, write
- Address Inputs: MUX
  - Register fields R1, R2 of instruction
  - Stack Pointer SP
  - Register RA (hard-wired, JMP)

Design Approval – By Week 6 in Your Lab

- **Concrete RTN for all 16 instructions**
- Functional Block Diagram
  - Components (chip number & function)
- Datapath along with control signals

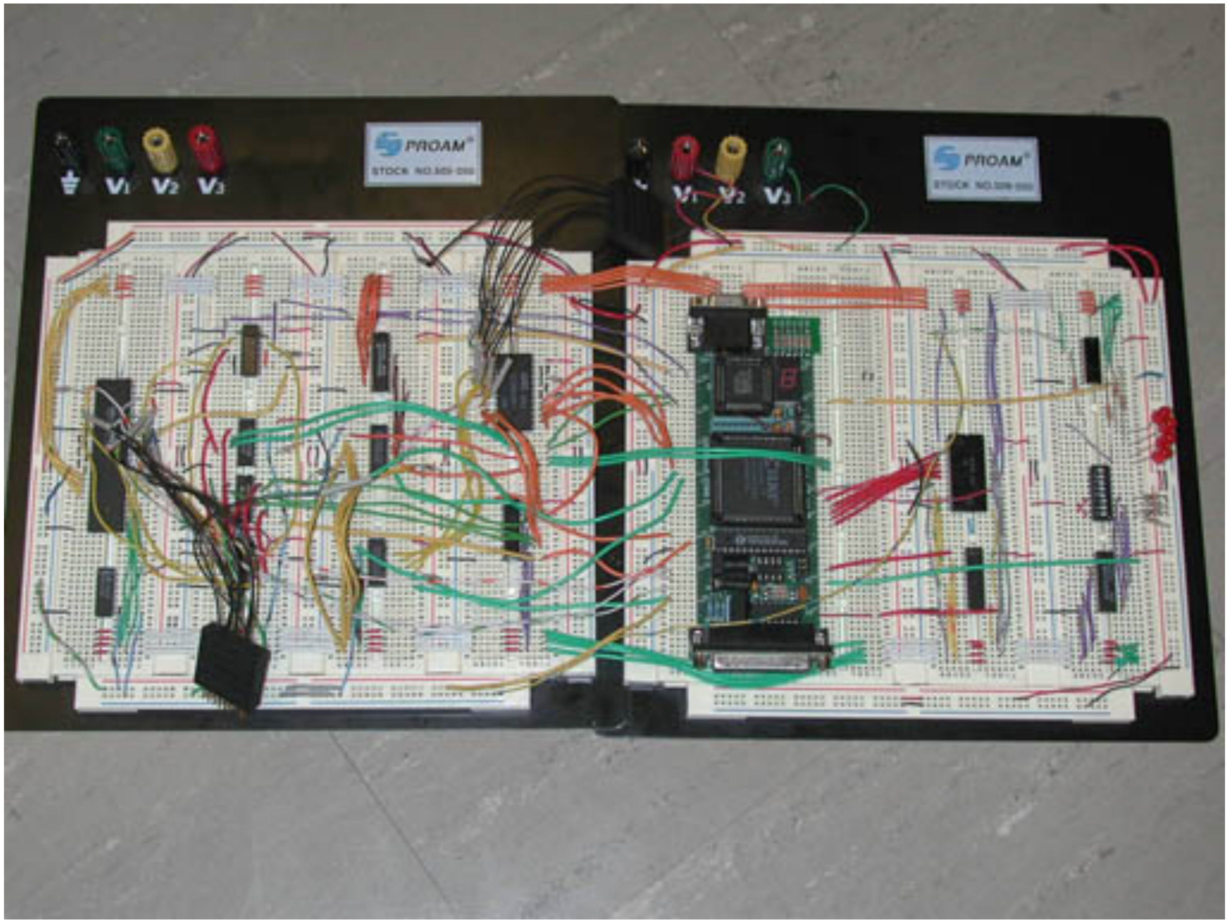
Block Diagram (Example)



ModelSim Simulation for BURP – By Week 7

- Verilog/VHDL for entire BURP (as close to your implementation as possible)
- Testbench for the test program
- Correct simulation results for desired behaviors/responses

Implementation and Debugging



Demo to TA by Week 9!