

## 1.0 FEATURES

- Fully self-contained 8 bit microprocessor IP, synthesizeable on most CPLD or FPGA with 11K gates or less.
- One integrated 8 bit quasi-bidirectional I/O port.
- Accumulator based architecture keeps code complexity low while minimizing gate count.
- Flexible external stack implementation with on board stack pointer.
- Flexible 25 instruction set with 3 addressing modes.
- Opcode length of one to three bytes
- Opcode execution cycle of two to six cycles.
- Total addressable code space of 4Kbyte and 4Kb of sram.

## 2.0 SYSTEM OVERVIEW

PC (PopCorn) is a complete 8 bit CISC microprocessor IP core. With the addition of 4Kbyte of code RAM and 4Kbyte of SRAM, a complete computer system can be implemented. As the name suggests, PC's architecture is a

relatively simple, with many similarities to Intel 8085. At the same time, the instruction set offers a very flexible programming potential. A mid-capacity CPLD or FPGA, such as Lattice 3256, with PLD gate count of 11K can be used to house the synthesized IP. Similarly, due to the modular design, multiple lower capacity devices can be used. Three modules comprises the PC core:

### PC.v

This is the highest level module. It serves as a wrapper of the other two modules, as well as providing two clock sources to the modules.

### POPCORN.v

This module is the datapath of the processor. It contains all of the onboard registers, ALU, program and stack pointers.

### SEQUENCER.v

This module contains the sequencing state machine responsible for the proper decode and execution of the opcodes.

Figure 1 illustrates the block diagram of PC as well as the inter-module signals.

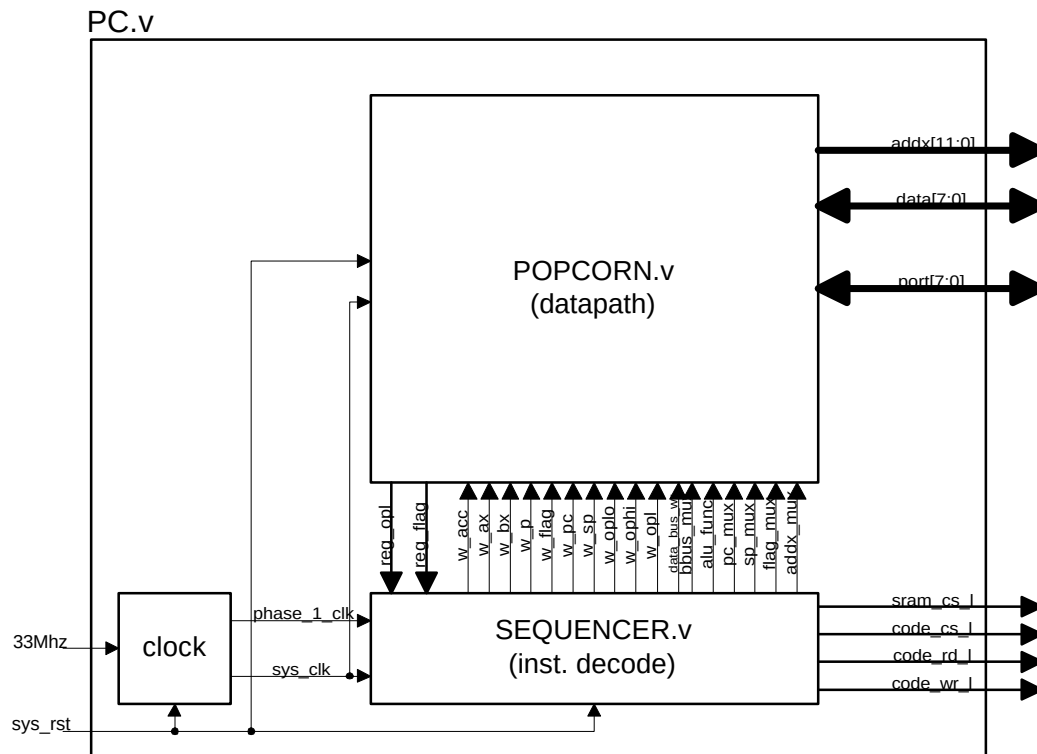


Figure 1 Modules and inter-module signals comprising PC

## TEST AND OBSERVABILITY

## TOOLS

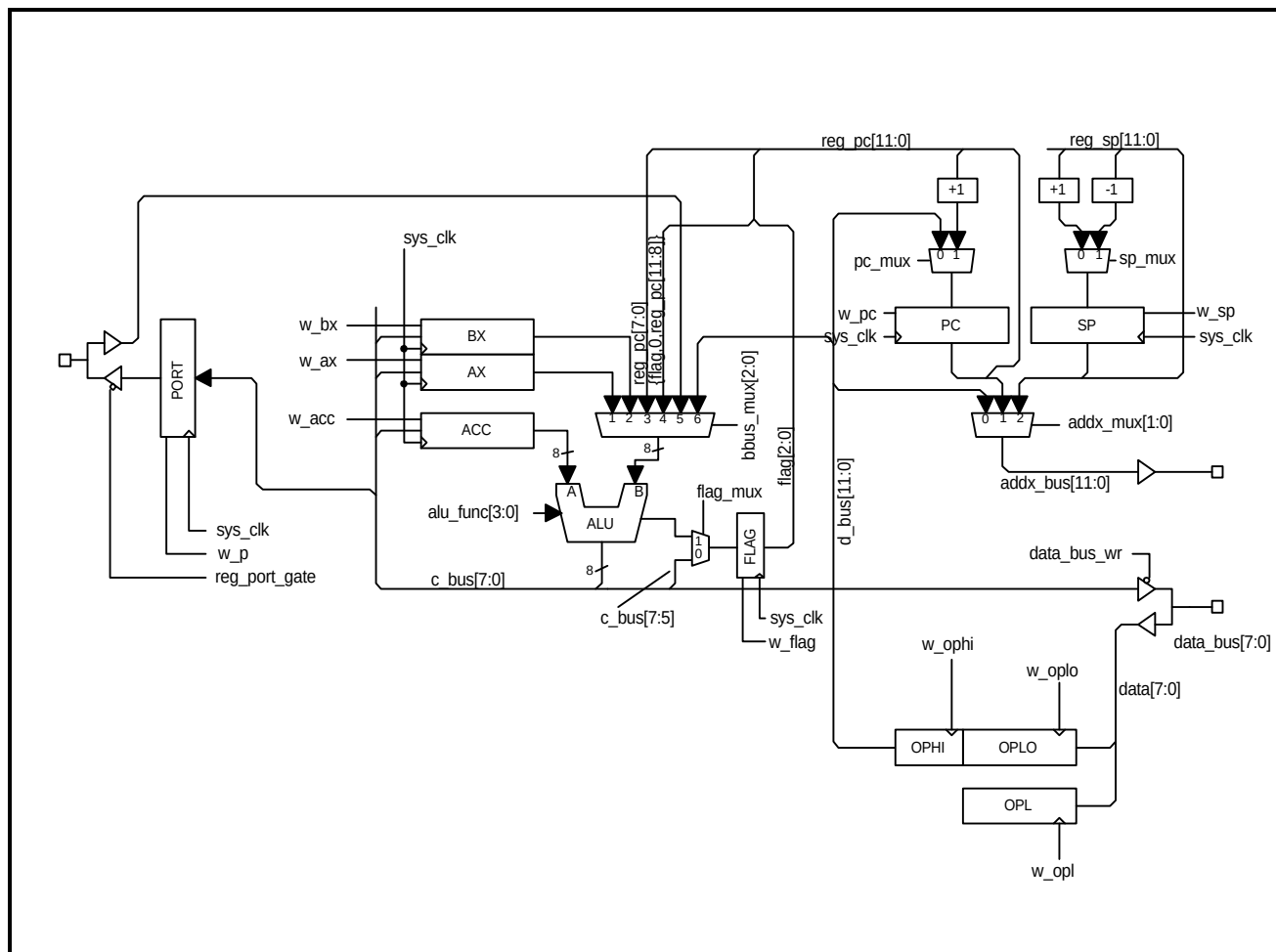


Figure 2 Popcorn.v Internal block diagram

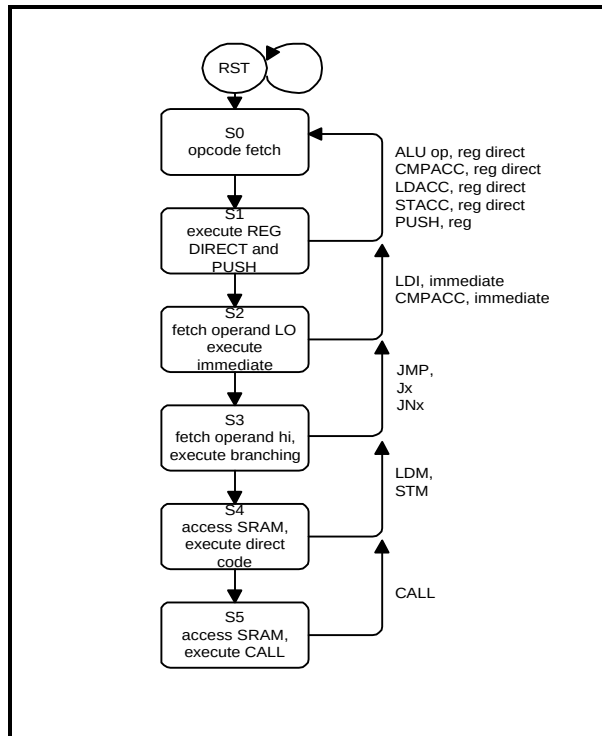


Figure 3 Sequencer state machine

| MNEMONIC | OPERAND | EXECUTION | MODE            | OPERATION                                                        | FLAGS |   |   |
|----------|---------|-----------|-----------------|------------------------------------------------------------------|-------|---|---|
|          |         |           |                 |                                                                  | CY    | P | Z |
| LDI      | 2       | 3         | immediate       | loads an immediate data into ACC, AX, BX or P register           | -     | - | - |
| LDACC    | 1       | 2         | register direct | loads the accumulator from AX or BX register                     | -     | - | - |
| STACC    | 1       | 2         | register direct | store the content of accumulator to AX to BX register            | -     | - | - |
| LDM      | 3       | 5         | direct          | loads the accumulator from content of memory address             | -     | - | - |
| STM      | 3       | 5         | direct          | stores the content of accumulator to memory location             | -     | - | - |
| ADDACCI  | 2       | 3         | immediate       | adds the accumulator with an immediate                           | √     | √ | √ |
| SUBACCI  | 2       | 3         | immediate       | subtracts the accumulator with an immediate                      | √     | √ | √ |
| ANDACCI  | 2       | 3         | immediate       | bitwise ANDs the accumulator with an immediate                   | √     | √ | √ |
| ORACCI   | 2       | 3         | immediate       | bitwise ORs the accumulator with an immediate                    | √     | √ | √ |
| XORACCI  | 2       | 3         | immediate       | bitwise XORs the accumulator with an immediate                   | √     | √ | √ |
| CMPACCI  | 2       | 3         | immediate       | compares the accumulator with an immediate                       | √     | √ | √ |
| ANDACC   | 1       | 2         | register direct | adds the accumulator with either AX, BX or P register            | √     | √ | √ |
| SUBACC   | 1       | 2         | register direct | subtracts the accumulator from AX,BX or P register               | √     | √ | √ |
| ANDACC   | 1       | 2         | register direct | bitwise ANDs the accumulator with AX, BX or P register           | √     | √ | √ |
| ORACC    | 1       | 2         | register direct | bitwise ORs the accumulator with AX,BX or P register             | √     | √ | √ |
| XORACC   | 1       | 2         | register direct | bitwise XORs the accumulator with AX,BX or P register            | √     | √ | √ |
| NOTACC   | 1       | 2         | register direct | performs one's complement on accumulator                         | √     | √ | √ |
| SHRACC   | 1       | 2         | register direct | logical shifts the accumulator right by 1 bit through carry      | √     | √ | √ |
| SHLACC   | 1       | 2         | register direct | logical shifts the accumulator left by 1 bit through carry       | √     | √ | √ |
| CMPACC   | 1       | 2         | register direct | compares the accumulator with AX,BX or P register                | √     | √ | √ |
| JE       | 3       | 4         | direct          | performs jump to address if Z (zero) flag is set                 | -     | - | - |
| JNE      | 3       | 4         | direct          | performs jump to address if Z (zero) flag is cleared             | -     | - | - |
| JP       | 3       | 4         | direct          | performs jump to address if P (positive) flag is set             | -     | - | - |
| JN       | 3       | 4         | direct          | performs jump to address if P (positive) flag is cleared         | -     | - | - |
| JC       | 3       | 4         | direct          | performs jump to address if CY (carry) flag is set               | -     | - | - |
| JNC      | 3       | 4         | direct          | performs jump to address if CY (carry) flag is cleared           | -     | - | - |
| JMP      | 3       | 4         | direct          | jumps unconditionally to address                                 | -     | - | - |
| CALL     | 3       | 6         | direct          | jumps to address but saves the return address and FLAGS to stack | -     | - | - |
| RET      | 3       | 3         | direct          | jumps to address saved to stack                                  | -     | - | - |
| PUSH     | 1       | 2         | direct          | saves ACC,AX,BX, FLAG or P register to stack                     | -     | - | - |
| POP      | 1       | 3         | direct          | restores ACC,AX,BX FLAG or P register from stack                 | -     | - | - |

## Opcode Format

| IMMEDIATE   | REGISTERED             | DIRECT                           | INDIRECT | IMMEDIATE OPCODE | REGISTERED OPCODE    | DIRECT OPCODE        | INDIRECT OPCODE |
|-------------|------------------------|----------------------------------|----------|------------------|----------------------|----------------------|-----------------|
| LDI reg, #n | LDACC reg<br>STACC reg | LDM reg, #addx<br>STM reg, #addx |          | 10ddd111         | 10001rrr<br>10010rrr | 01100ddd<br>01101ddd |                 |
| ADDACCI #n  | ADDACC reg             |                                  |          | 00000111         | 00000rrr             |                      |                 |
| SUBACCI #n  | SUBACC reg             |                                  |          | 00001111         | 00001rrr             |                      |                 |
| ANDACCI #n  | ANDACC reg             |                                  |          | 00010111         | 00010rrr             |                      |                 |
| ORACCI #n   | ORACC reg              |                                  |          | 00011111         | 00011rrr             |                      |                 |
| XORACCI #n  | XORACC reg             |                                  |          | 00100111         | 00100rrr             |                      |                 |
|             | NOTACC                 |                                  |          |                  | 00101xxx             |                      |                 |
|             | SHRACC                 |                                  |          |                  | 00110xxx             |                      |                 |
|             | SHLACC                 |                                  |          |                  | 00111xxx             |                      |                 |
| CMPACCI #n  | CMPACC reg             |                                  |          | 01000111         | 01000rrr             |                      |                 |
|             |                        | JE #addx                         |          |                  |                      | 10011000             |                 |
|             |                        | JNE #addx                        |          |                  |                      | 10100000             |                 |
|             |                        | JP #addx                         |          |                  |                      | 10101000             |                 |
|             |                        | JN #addx                         |          |                  |                      | 10110000             |                 |
|             |                        | JC #addx                         |          |                  |                      | 10111000             |                 |
|             |                        | JNC #addx                        |          |                  |                      | 11000000             |                 |
|             |                        | JMP #addx                        |          |                  |                      | 11001000             |                 |
|             |                        | CALL #addx                       |          |                  |                      | 11010110             |                 |
|             |                        | RET                              |          |                  |                      | 11011110             |                 |
|             |                        |                                  | PUSH reg |                  |                      |                      | 01110ddd        |
|             |                        |                                  | POP reg  |                  |                      |                      | 01111ddd        |

## Register Mapping

| value | rrr      | ddd      |
|-------|----------|----------|
| 000   | -        | ACC      |
| 001   | AX       | AX       |
| 010   | BX       | BX       |
| 011   | -        | -        |
| 100   | FLAGS    | FLAGS    |
| 101   | PORT (P) | PORT (P) |
| 110   | -        | -        |
| 111   | -        | -        |