Parallax Propeller

by Kyle Kurzhal

1. Background/Contributions
   1. Parallax Inc. was founded in 1987 by Chip Gracey and Lance Walley
   2. In 1992, Parallax created the BASIC Stamp 1 (great for learning and hobbyists)
   3. Then, based on the BASIC Stamp model, the Propeller microcontroller was developed in 2006
   4. Parallax has been greatly involved in electronics education
      1. Stamps in Class program in 1997
      2. Released source code of executable portions of ROM (2008)
      3. Parallax Propeller 1 released as open source hardware under the GNU v3.0 (2014)
2. General Facts/Unusual Features
   1. Up to 80 MHz with low power usage
   2. 8 processors, each called a “Cog”
      1. Up to 20 MIPS
      2. Independent from other Cogs, allowing parallelization
   3. Centrally shared memory, called the “Hub”
      1. Contains RAM/ROM
      2. Mutually exclusive – only accessible in round-robin fashion to each Cog for two cycles
      3. Runs at half the system clock speed
   4. Timing of instruction executions is extremely important
3. Memory specifications
   1. Cog RAM/registers
      1. 2 KB each
      2. Long addressable
      3. Addresses $000-$1FF
   2. Main/Hub RAM/ROM
      1. 32 KB apiece
      2. Byte (8-bit), Word (2-byte), Long (2-word) addressable
      3. Addresses $0000-$7FFF (RAM) and $8000-$BFFF (ROM)
      4. ROM contains character definitions, Log and Anti-Log tables, a Sine table, a boot loader, and the Spin interpreter
   3. Data stored in Little Endian format
4. Registers
   1. 512, 32-bit registers per Cog
      1. 496 general purpose registers (addresses $000-$1EF)
      2. 16 special purpose registers (addresses $1F0-$1FF)
   2. Registers are also considered to be RAM
      1. May store entire instructions, data values, or addresses
5. Data Types
   1. Binary (denoted with a beginning “%”)
   2. Quaternary (denoted with a beginning “%%”)
   3. Decimal integers
   4. Decimal floating-points (denoted with “e” for scientific notation or with “.” for decimals)
   5. Hexadecimal (denoted with a beginning “$”)
   6. Characters
6. Instruction Set
   1. RISC-based architecture
   2. Two main languages
      1. Spin (interpreted from Hub RAM)
      2. Propeller Assembly (stored directly in Cog registers)
   3. Instructions
      1. Logic
      2. Arithmetic
      3. Control
      4. I/O
      5. Hub
7. Instruction format
   1. 32-bit instructions
   2. Propeller Assembly
      1. Instruction - bits 31-26
      2. Effects (flags) - bits 25-22
      3. Execution condition (If statement) - bits 21-18
      4. Destination register - bits 17-9
      5. Source register or 9-bit immediate value - bits 8-0
   3. Spin
      1. Interpreter stored in registers
      2. Execution instructions fetched from Hub RAM
8. Addressing modes
   1. Spin
      1. Direct addressing
      2. Indirect addressing
      3. Indexed addressing for special purpose register array (SPR)
      4. A form of base-relative addressing for certain instructions
   2. Propeller Assembly
      1. Direct addressing
      2. Indirect addressing
      3. Immediate addressing – denoted by '#' in source register
9. I/O
   1. 32 I/O pins
      1. 28 general purpose
      2. 4 special purpose (until after boot-up)
      3. May be simultaneously accessible
   2. Direction register – controls if a pin is considered output and if the associated Cog may output to it.
   3. INA register – input register
   4. OUTA register – output register
   5. Dependent upon the programmer