Super IO

Ross Glashan
James Wnorowski
6.270 : The Handyboard

- Created by Fred Martin (Media Lab) in 1995
- 2MHz 68HC11 + 32K RAM
- 6 Motors, 6 Servos
- 8 Digital IO, 24 Analog IO
- 2 Shaft Encoders
- LCD, pushbuttons
- All IO controlled from software
Problems

- Handyboard is 10 year old design (68HC11 no longer available)
- No significant updates to design since 1999
- Slow, underpowered
- Everything done in software – heavy loading causes problems
- Contestants beginning to push HB beyond its capabilities
- Want new features
Proposed Solution

- New board design based around FPGA + Microcontroller
- Microcontroller performs high-level control
- FPGA performs low-level control (controls motors, IO ...)
- FPGA ("SuperIO Chip") - MCU communications via I2C
- All the features of the Handyboard and more
- Built-in closed loop motor control
- Automatic polling of sensor ports
- Graphics LCD controller
High Level Design
SuperIO Design
Interface I2C

- Microcontroller
  - set MOTOR1FEEDBACK = 2
  - OK
  - set MOTOR1VEL = 10
  - OK
  - get ENC1VEL
  - ENC1VEL = 10
  - get ANALOG7
  - ANALOG7 = 42

- SuperIO
Addressing Modules

- Each module has a base address + N registers
- Registers accessible over bus
- Address space allocated to each module type
  - 32 regs for motors
  - 24 for analog
  - ...


Example Driver Modules

Motor Module
- velocity (rw) [addr+0]
- direction (rw) [addr+1]
- current (r) [addr+2]
- feedback (rw) [addr+3]

Driver PWM

Encoder Module
- velocity (rw) [addr+0]
- position (rw) [addr+1]

Counter FSM

Analog Module
- control (rw) [addr+0]
- port (rw) [addr+1]
- result (r) [addr+2]

ADC FSM

LCD Module
- command (rw) [addr+0]
- data (rw) [addr+1]

Display FSM