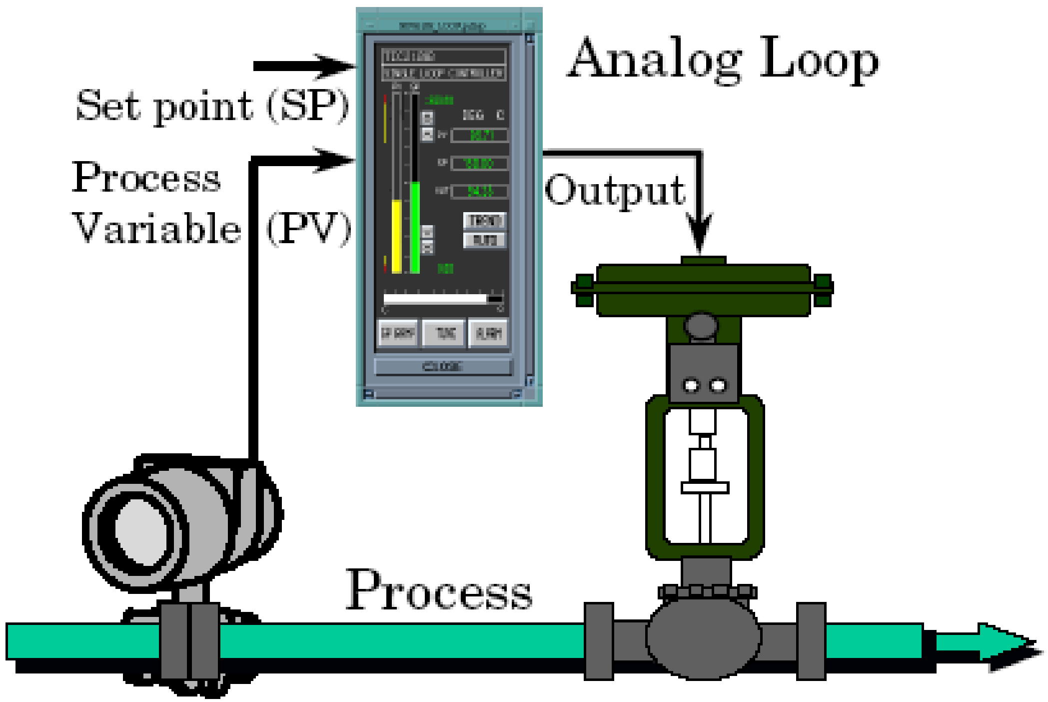
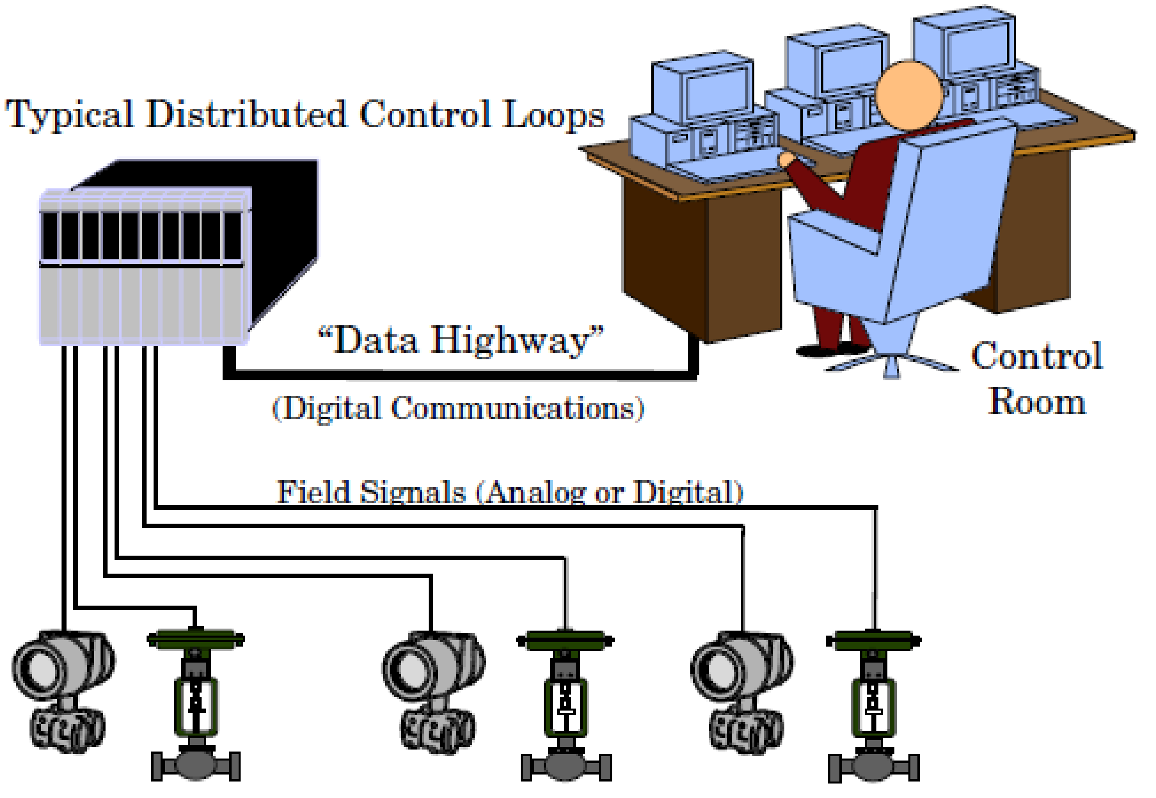
Module_3
DCS Controller

Simple process loop using a traditional single loop controller



Several loops share the same digital controller

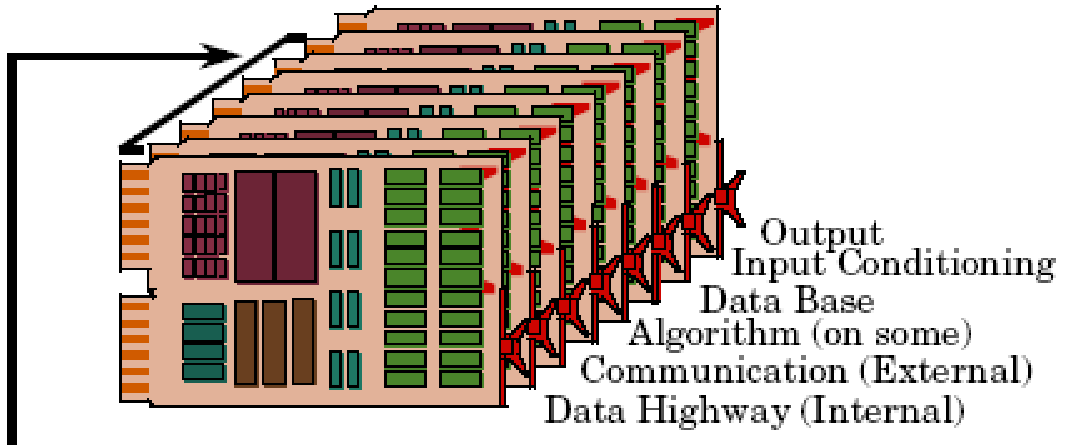


Architectures of Controllers

Two very general controller architectures emerged since 1990 and both types influenced the approach used in the newest of designs, namely:

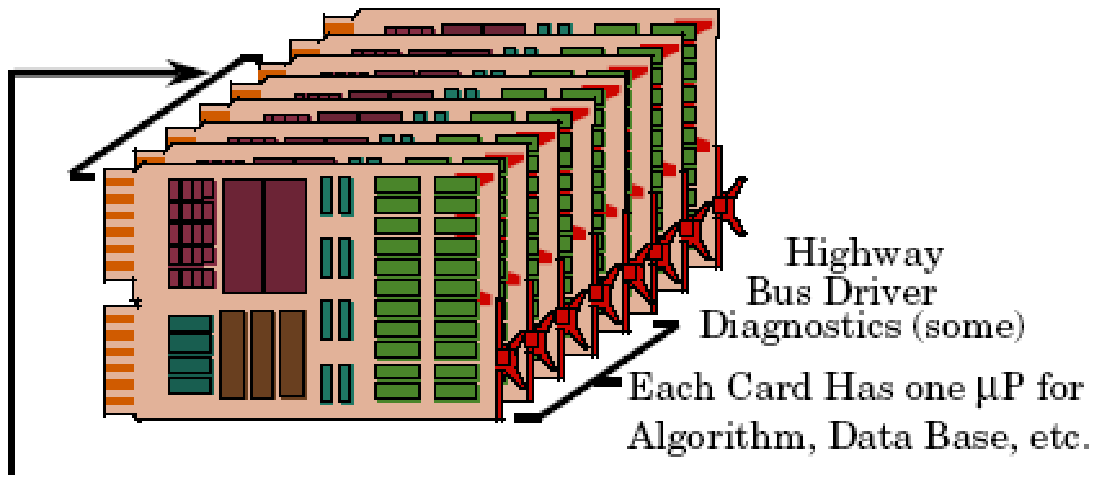
- Shared-function.
- Individual cards for each “control loop”

Shared-function Controller Rack



- **All "control loops" share several cards**
- Same sets for all
- Function processing is distributed:
  - ● I/O ● Control ● Data highway ● Communication

Individual cards for each loop set



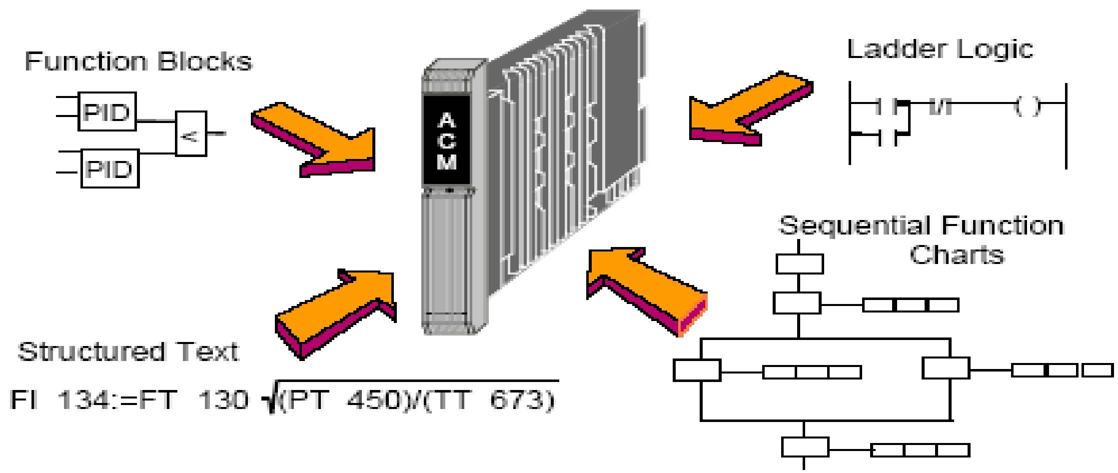
- **Individual cards for each "control loop"**
- Different sets per order
- Select card for each needed function:
  - Loop ● Logic ● Data Acquisition ● Multifunction

Multifunctiom Controller

- Later designs began to combine these ideas taking advantage of ever increasing processor power and memory capacities.
- Emerging from both of these earlier designs is a more current design in which all the functions are embedded on a single card, or module.
- Most architectures since the mid-1990s use what are called multifunction controllers rather than loop controllers, logic controllers, and separate application controllers. This approach also provides the advantages of the single hardware/software set.

Multifunctiom Controller

- Single Control Module with Multiple "Languages" (and Capabilities) Within the Same Module

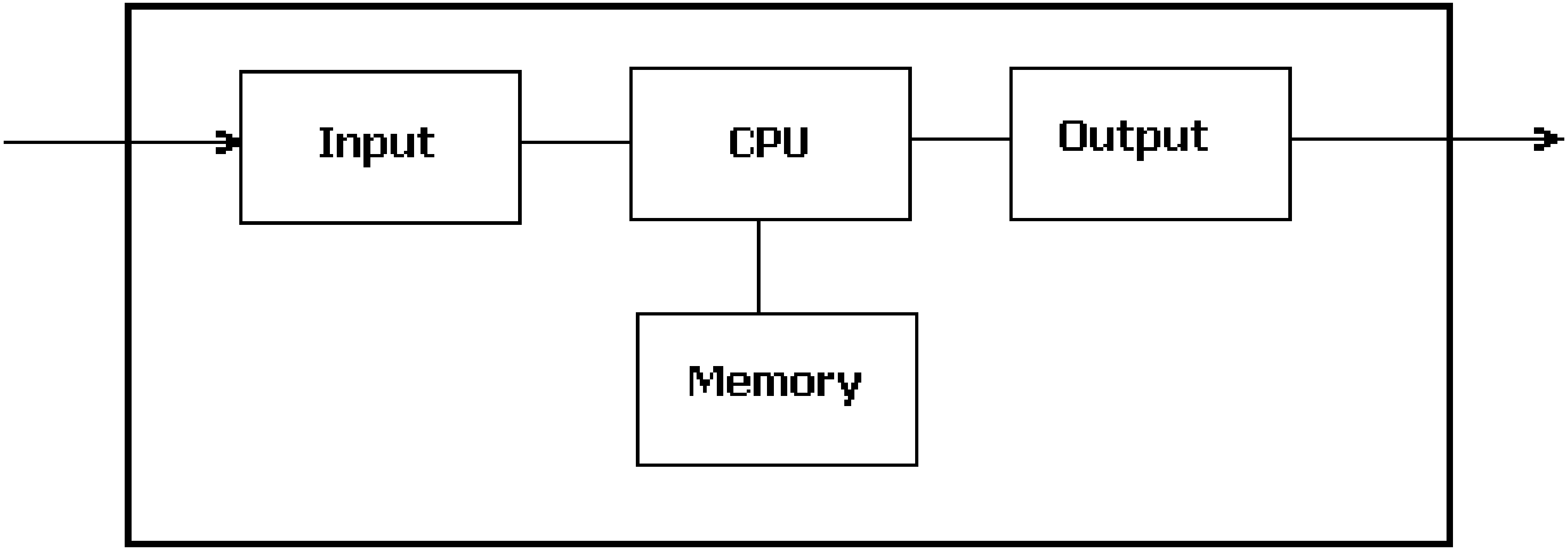


- Continuous, discrete, sequential control
- Four blended languages (IEC 1131-3)

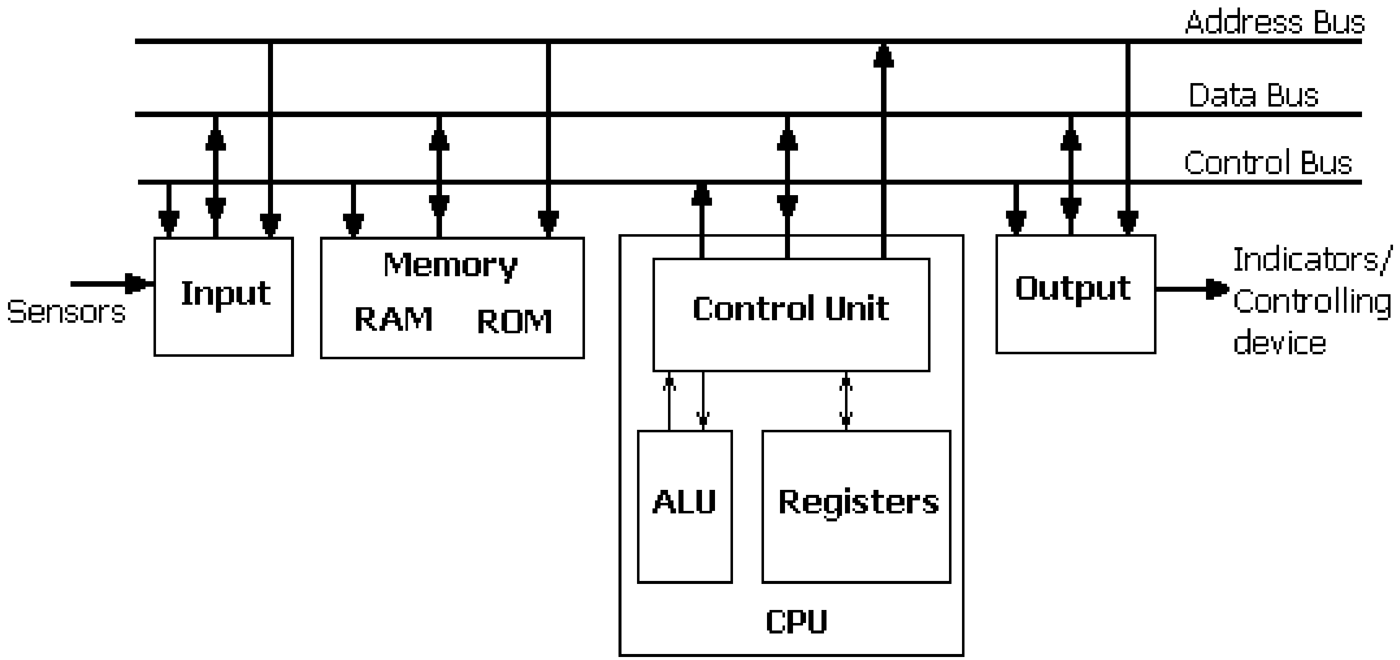
Multifunctiom Controller

- The advantages of having multiple loops residing within the same card makes it even more possible to create some very powerful multiloop control strategies.
- This capability defeats the purpose of “single loop integrity,” which is near impossible to achieve with any kind of interlocking control strategy.
- The only protection for today’s control strategies is redundant controllers, which are now more practical and reasonably priced than before.

Block Diagram of controller



Microprocessor based Controller



Controller Software Structures

Controller Software Structures

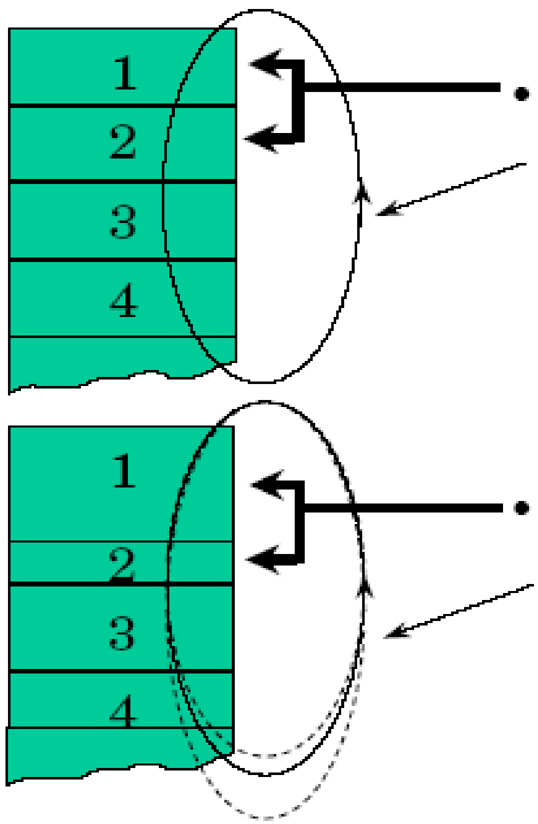
- Programming
  - Firmware is software that is burned into a PROM and always stays the same, so that certain routines are always done the same way
  - Software is the set of instructions in the computer that tell it how to function.

Organizing execution time for control action

- chosen. Just as hardware designs have different appearances, so do software designs even though the function may be the same.
- The time the processor takes to go through all the lines of code is called scan time.
- Most control vendors incorporate real-time execution of loop functions within their design.

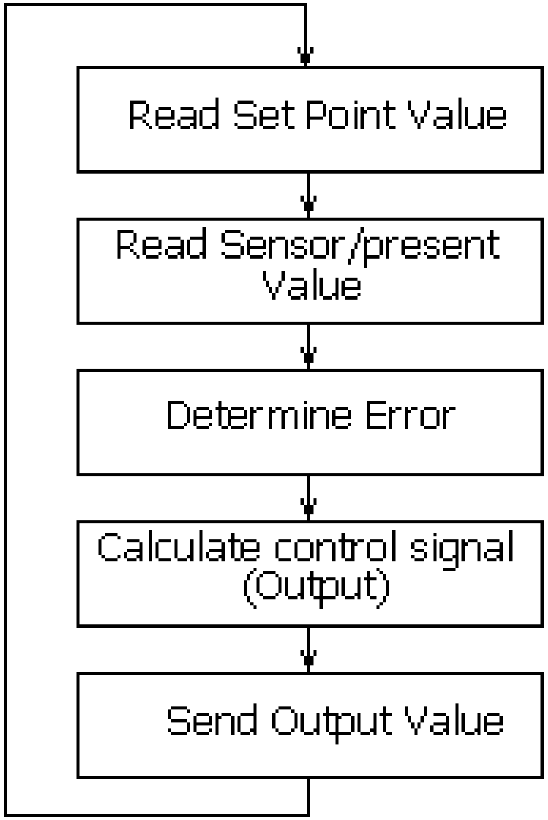
Organizing Execution Time for Control Action

Fixed Times for Flexibility vs. Variable Times for Capacity

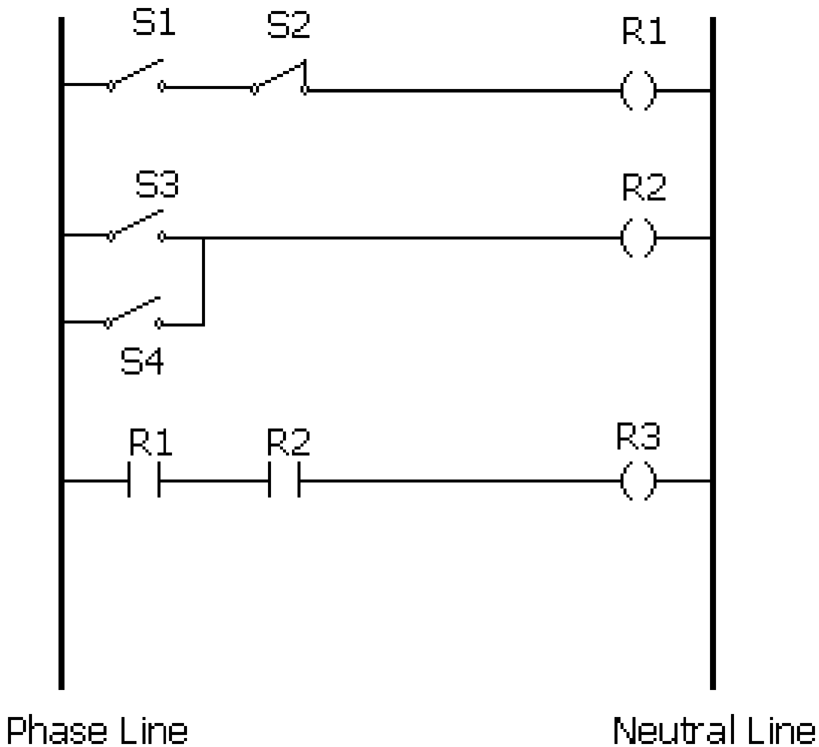


- Fixed “time slots”
  - Fixed scan times
  - Easier to configure and re-configure!
  - Typically has algorithm card
  - Functions “reused” during scan
- Variable “time slots”
  - Scan time varies w/ density
  - Can pack in more functions
  - Algorithms are packaged lines of programs which are used once per scan

Typical Control Program



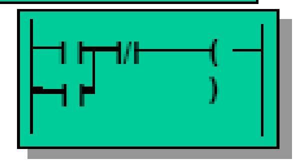
Typical Ladder logic Program



Programming vs. Configuration

```
10110110
00000100
00101100
```

```
LOOP1  LDAA  TEMP
       SUBA  #150
       blt   LOOP1
       LDAA  #$FF
       STAA  MOTOR
```



AND, OR, NOR,...

$[Y=(X_1+X_2+X_3+X_4+X_5)/5]$

- Machine code →
- Assembly language →
- Relay ladder logic →
  Boolean logic →
- "High" level →
  - Fortran – Cobol – Basic - C++ – Visual Basic
- Algorithms & soft wiring
  - Menu entry – Fill-in-the-blanks – Graphic blocks
  - Object oriented technology – IEC 1131-3 standard
  - Graphical configuration

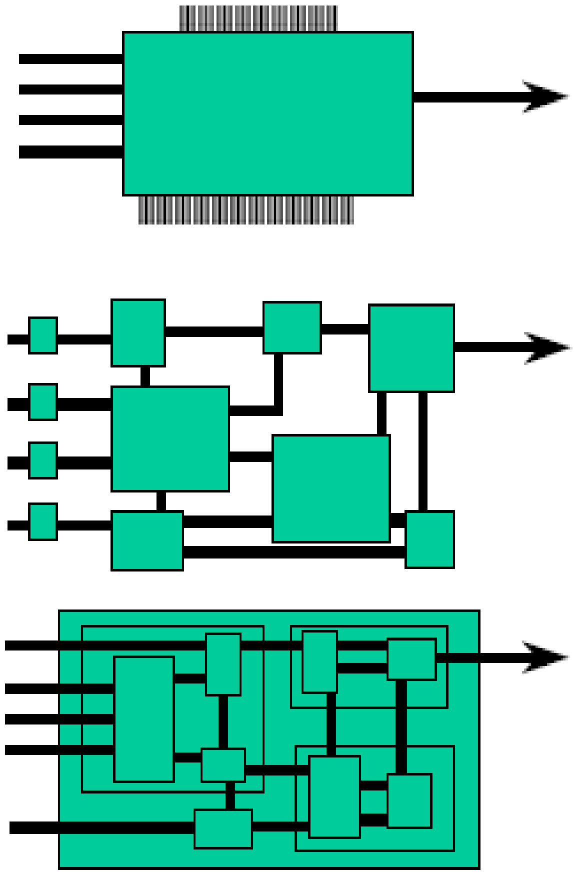
Growing From Programming to Configuration: Advances in Software Structures

Function Blocks

- To overcome the difficulty of software languages for users, vendors develop function blocks to arrange” control strategies by connecting them in various ways with “soft wiring.”

- Some vendors use large number of small and flexible function blocks and others use much less number of a larger and complete & powerful function blocks.

Function blocks grow to object oriented configuration



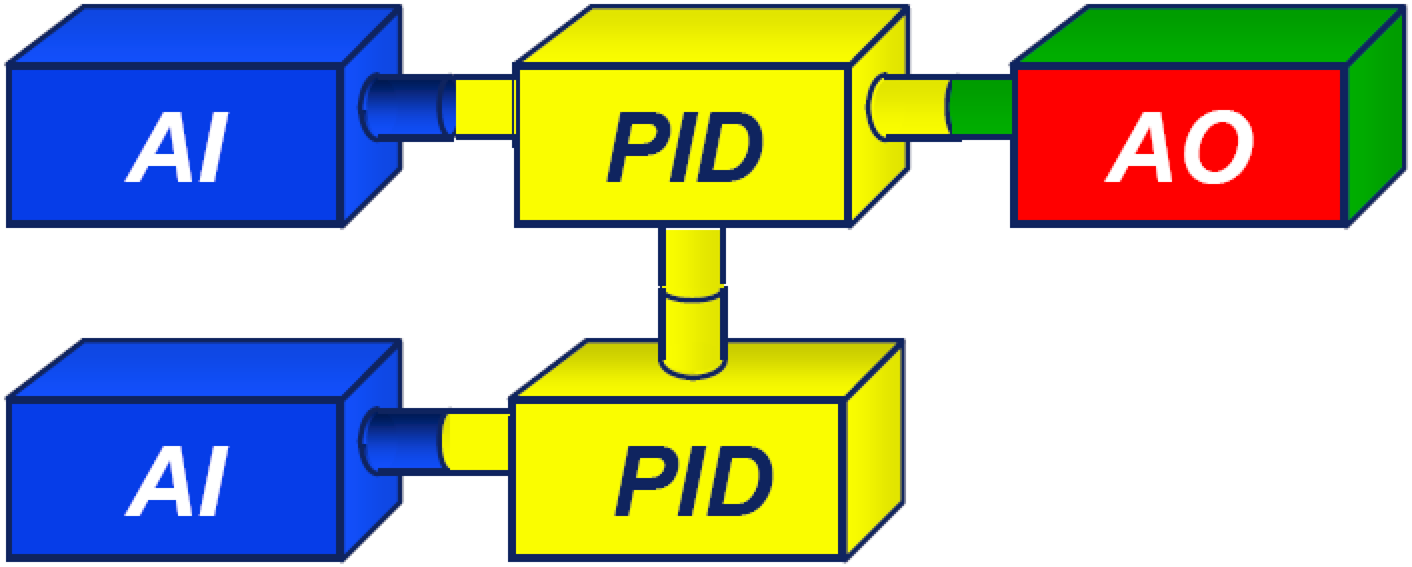
- Fewer large blocks
  - Many features in each
  - Easy re-configuration
- Many small blocks
  - More flexible
  - Complex to create or change configuration
- Object oriented
  - Build layers of large blocks from smaller
  - Flexible but simple

IEC61804 standard defines function block architecture for the process industry

- The Foundation Fieldbus profile is used by many manufacturers of fieldbus devices

- Many control systems have adopted the Foundation Fieldbus function block architecture and specific function blocks

Function blocks



Basic FF function blocks

- Discrete Input
- Discrete Output
- Analog input
- Analog Output
- PID, PI, I Controller
- P, PD Controller
- Control Selector
- Manual Loader
- Bias/Gain Station
- Ratio Station

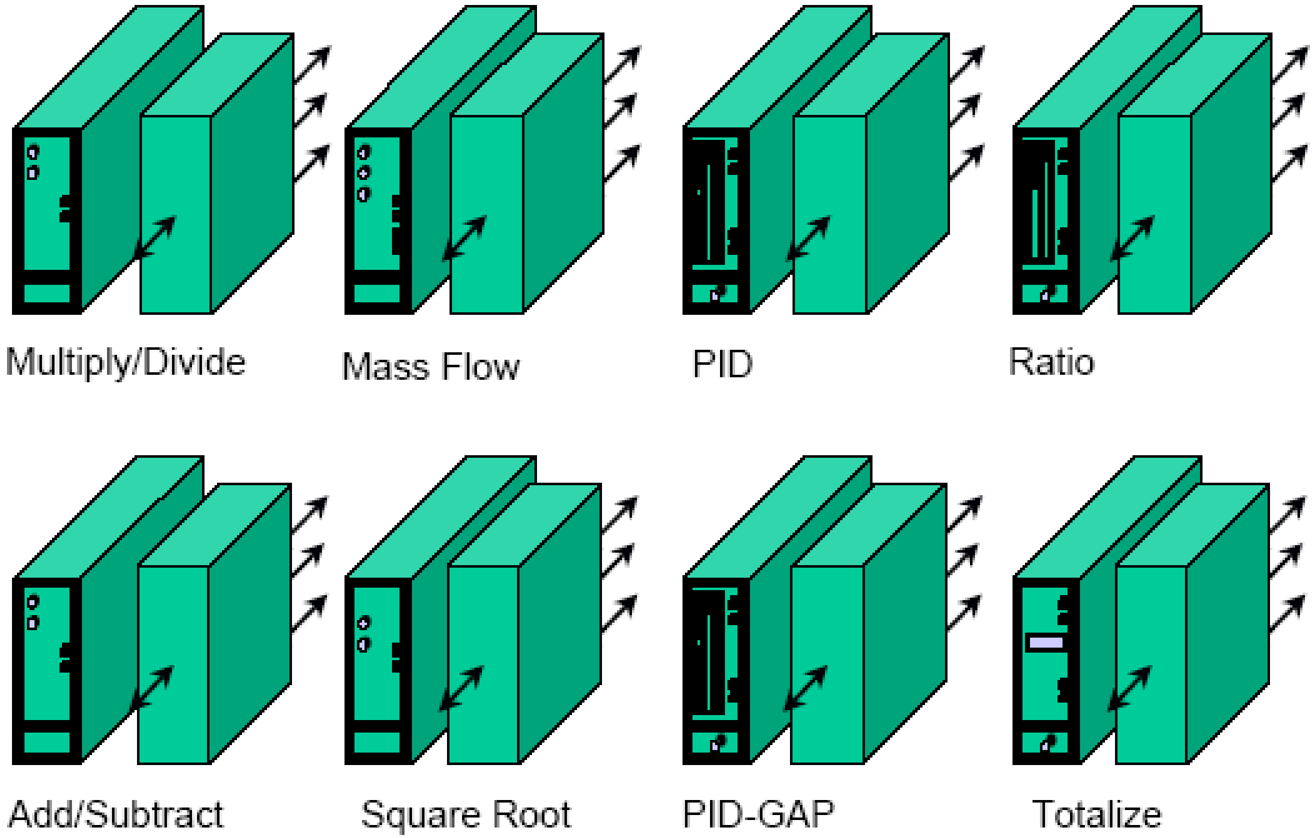
Advanced FF function blocks

- Pulse Input
- Complex Analog Output
- Complex Discrete Output
- Step Output PID
- Device Control
- Setpoint Ramp Generator
- Splitter
- Input Selector
- Signal Characterizer
- Lead Lag
- Deadtime
- Arithmetic
- Calculate
- Integrator(Totalizer)
- Timer
- Analog Alarm
- Discrete Alarm
- Analog Human Interface
- Discrete Human Interface

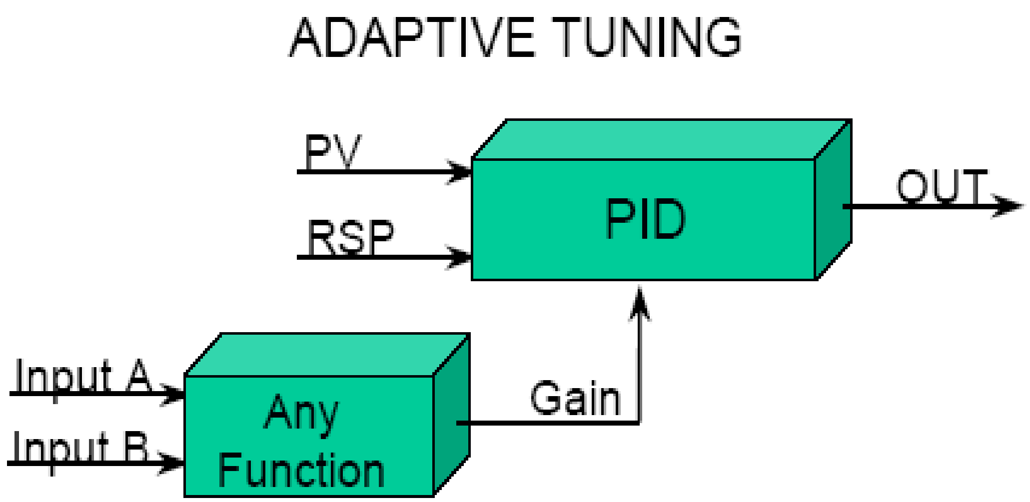
Object-Oriented

- The object-oriented configuration allows you to build layers of larger blocks out of smaller blocks.
- This means you can assemble the exact type of three-mode controller you like, with whatever alarms and status conditions you wish it to report. Then you save it with the identity of a single unit (large block).
- You can then build very complex configurations of control strategies without having to build all the details each time.

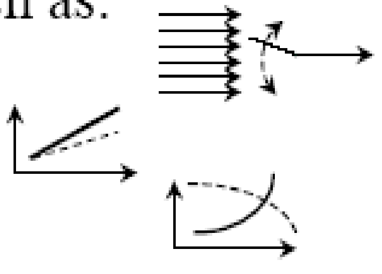
Hardware to Firmware: function blocks can be “instruments”



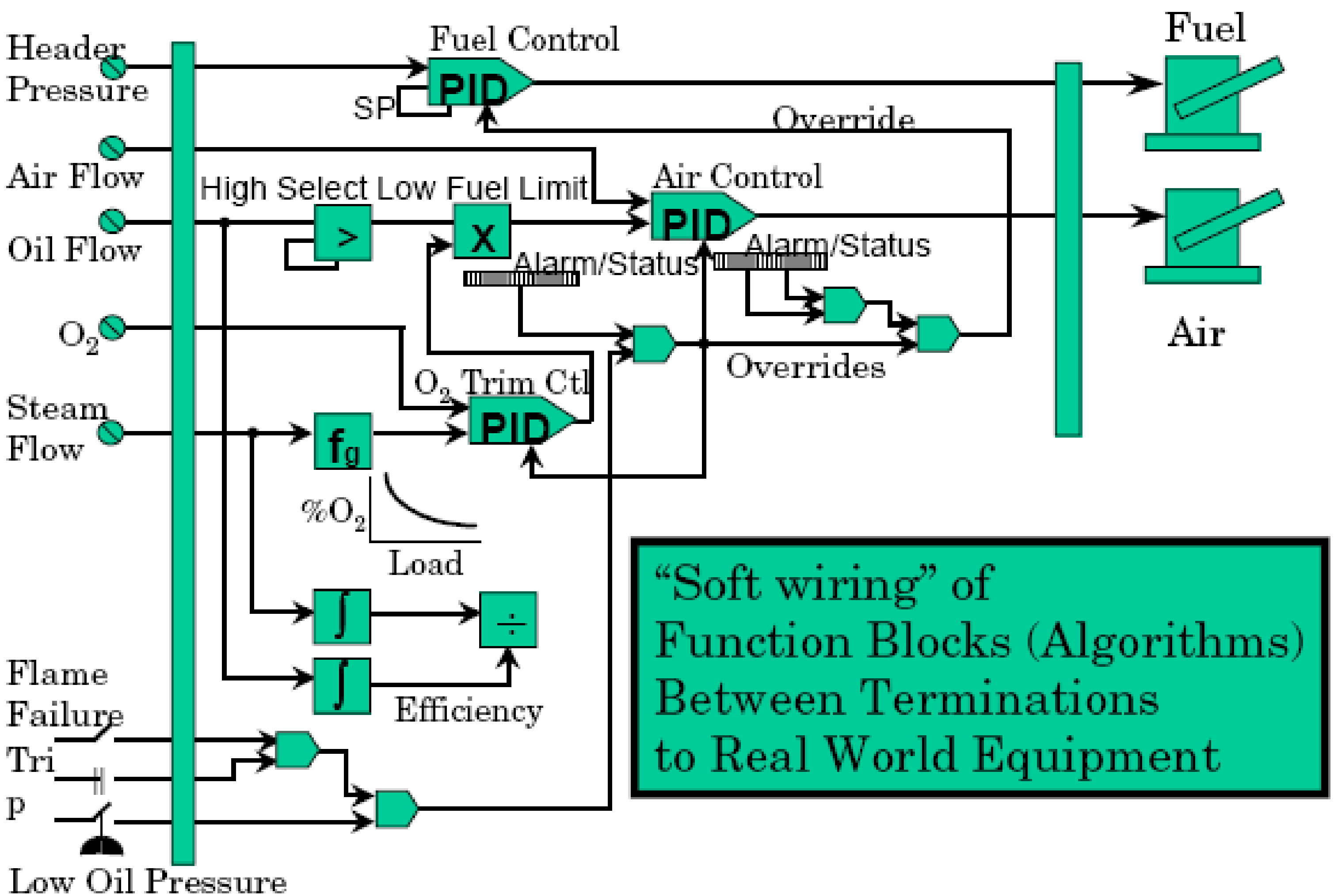
Combining function blocks for powerful control strategies



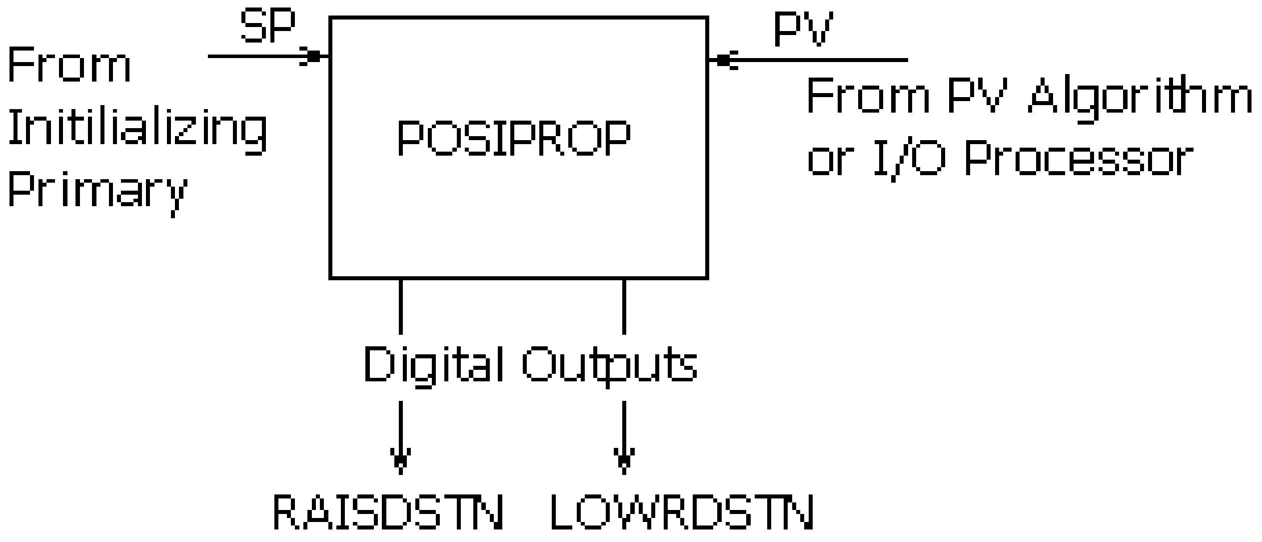
- PID gain can be continuously changed by any other function, such as:
  - Multiposition Switch
  - Ramp generator
  - Function generator



Connecting Function Blocks Through "Soft Wiring"



Position-Proportional Control



e.g. driving motorised valves

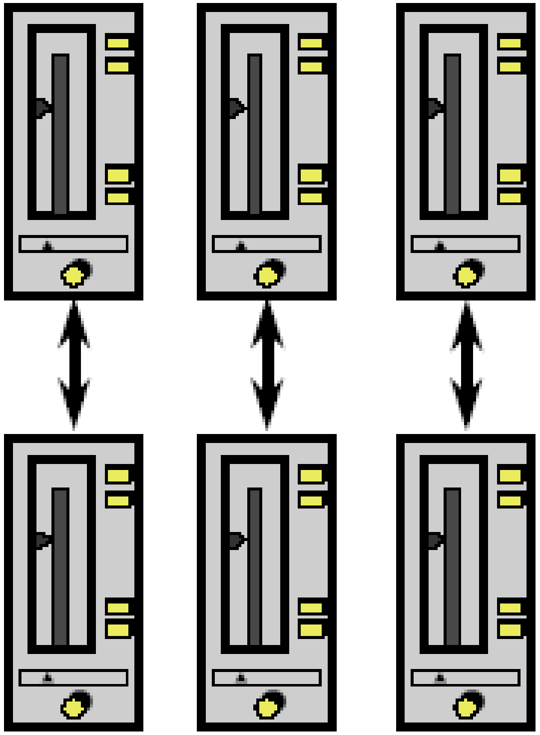
DCS configuration

- Type of control points/slots
- Frequency of execution
  - Each function requires 'Processing Units'
  - Complex controls take more processing time
  - Limit scan rate

Controller Redundancy

Controller Redundancy

Redundant Single Loop Control for "Single Loop Integrity"

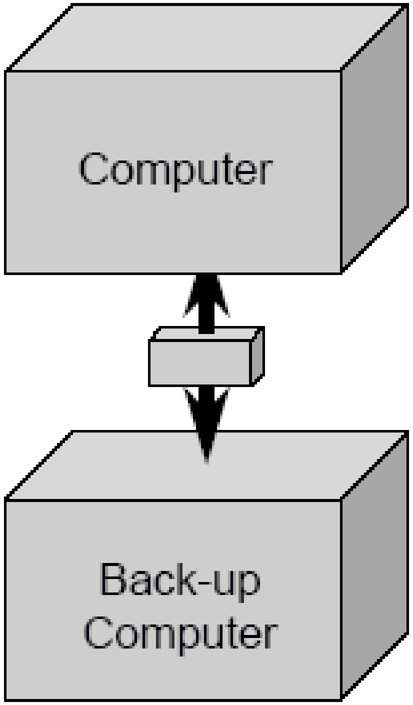


- Difficult if any control strategy is involved

Consider:

One Loop
vs.
Interconnected
Loops

Mainframe Computer Redundancy

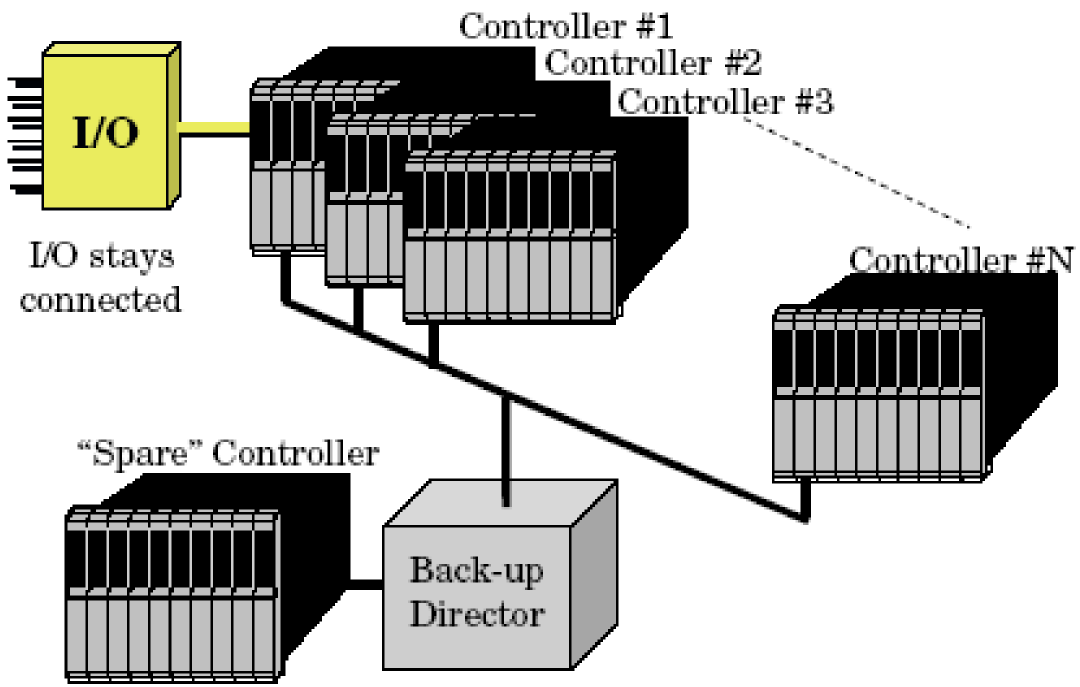


- Complete strategy
  ...but:
- Care needed to duplicate exact program in back-up computer
- Switch over
  - often took time,
  - was complex

Mainframe Computer Redundancy

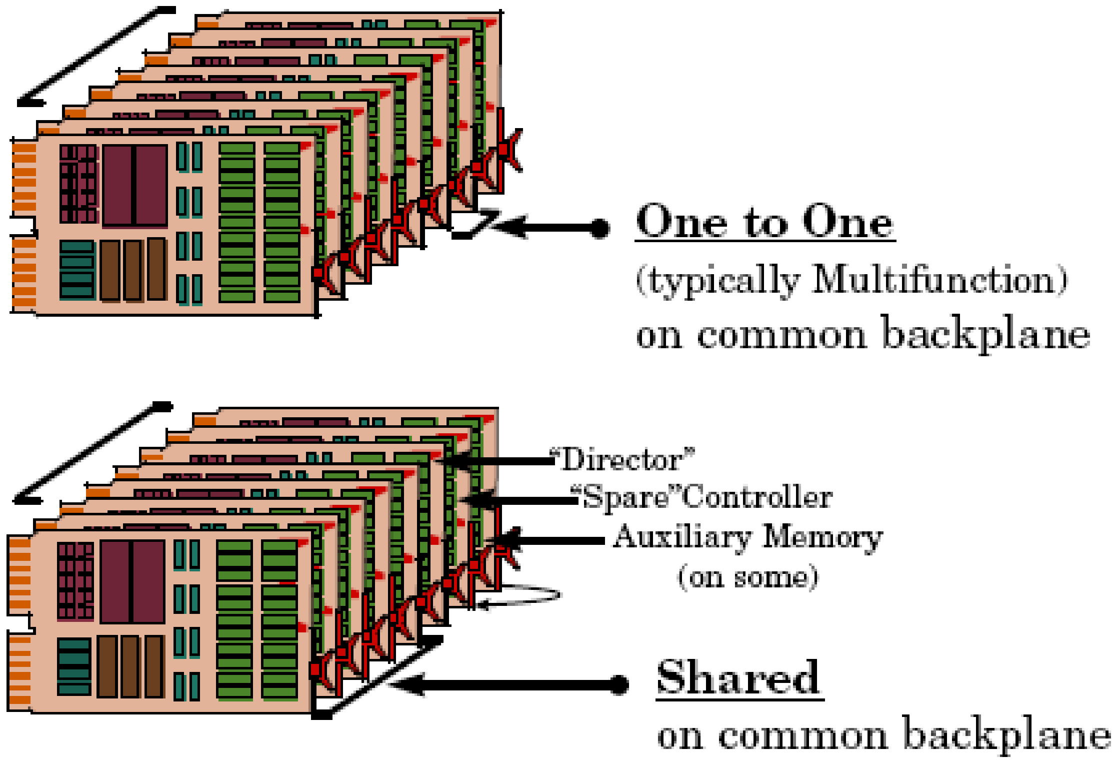
- Very difficult to make but it was at least possible.
- A second computer had to be programmed exactly like the first.
- When a failure occurred in the first , the switchover routine had to compare each line of code before allowing the change which would take sometimes hours. This would be very unacceptable in most process control applications.

Shared Loop Redundancy Among Controllers

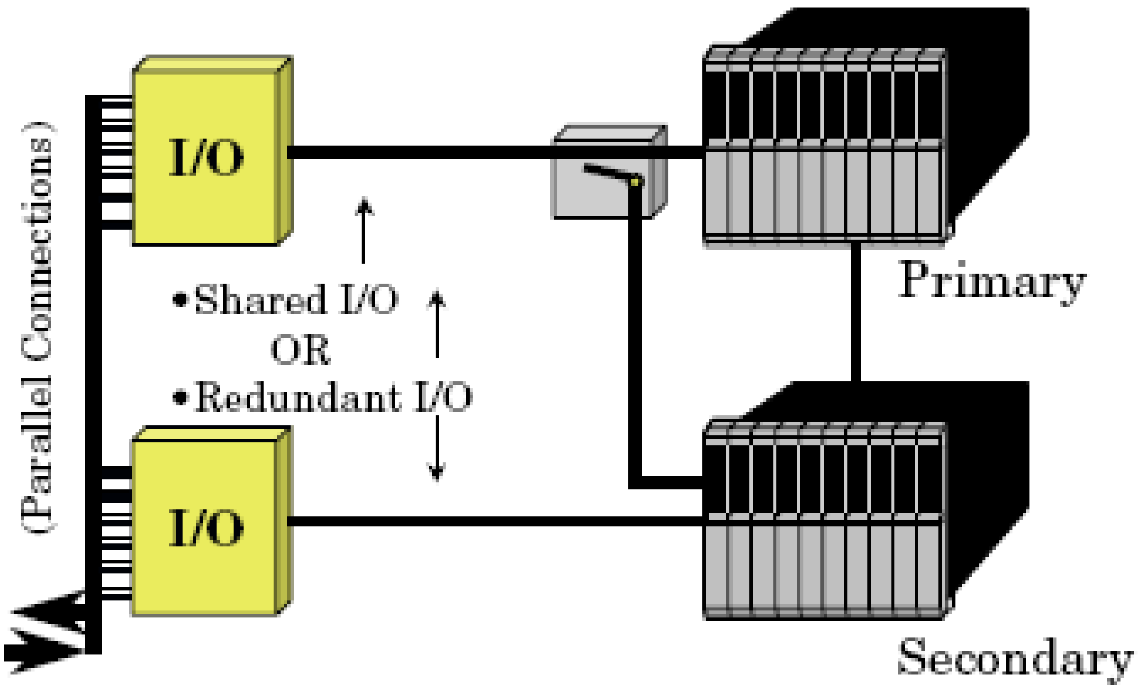


- "Shared card" version
- Entire strategy is backed up by "spare"

Microprocessor-based redundancy in “single card” controller racks



One-to-one redundant racks, I/O, even power



- "Shared card" version; some module versions
- Entire strategy backed

Redundancy techniques

- Both types (shared and one-to-one) have redundant cards on the same backplane, some use redundant circuit pathways on that backplane.
- Some vendors use redundant bus paths on the backplane to assure security.
- In those “single card” controller racks , only “multifunction cards” can back up their identical twin.

Redundancy techniques

- Most, but not all, will automatically refresh with any changes that occur in the primary controller. Some vendors permit the backup to be loaded with a different configuration and call that a benefit, namely, to allow “live reconfiguration.”

- Some vendors have designed the redundant controller as a “hot spare” to continually track the current values as well as any online or off line changes in the configuration in the primary controller.

Connections to the Controller

Connections to the Controller

- The two main sources of inputs and outputs to the controller are the interface with the process, and the interface with the operator.
- Interface with the process manage signals to and from the process.
- The interface with the human operator involve how to present information from the controller.

Connections to the Controller

- Inputs and outputs to a controller traditionally have been analog or discrete switching. Some pulse inputs have been available over the years for devices such as frequency measurements, and pulse outputs have been available for certain motor drive units.

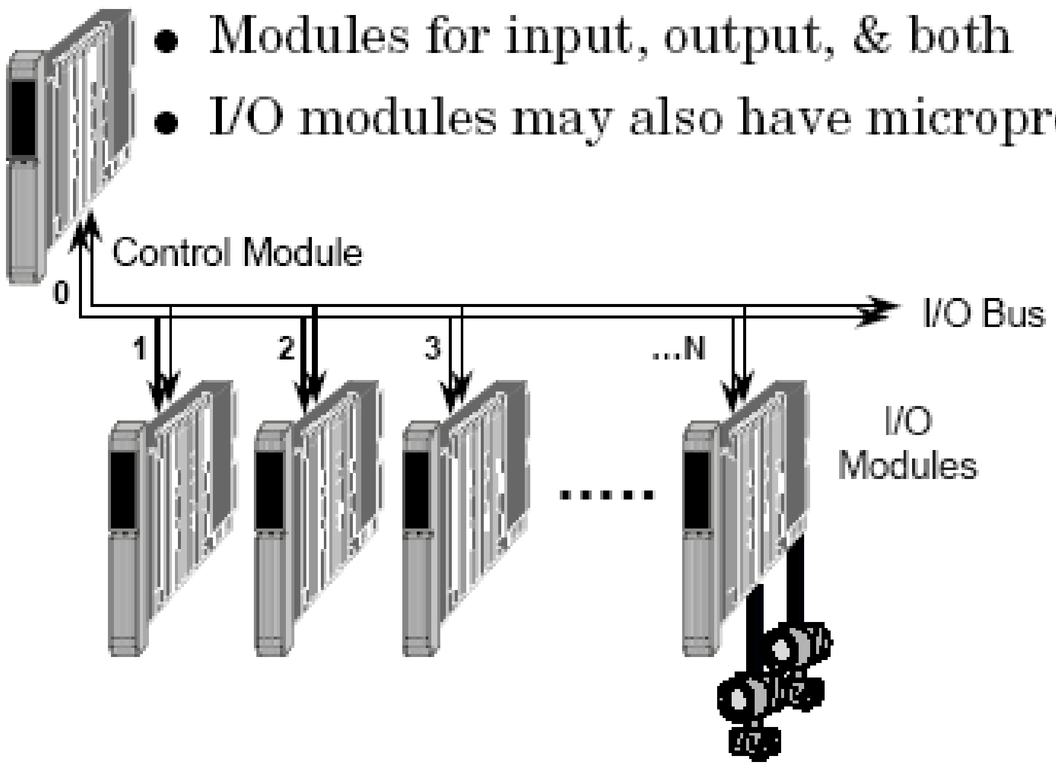
- Connected to most controller modules today are many versions of input/output (I/O) modules.

- Rather than using the controllers, the I/O modules themselves more and more contain microprocessors for conversion from analog to digital (and back) and for linearization and signal conditioning.

Input/output Subnetworks to Field Devices

Even Local I/O Comes In on a Bus Within the Controller "Card Cage"

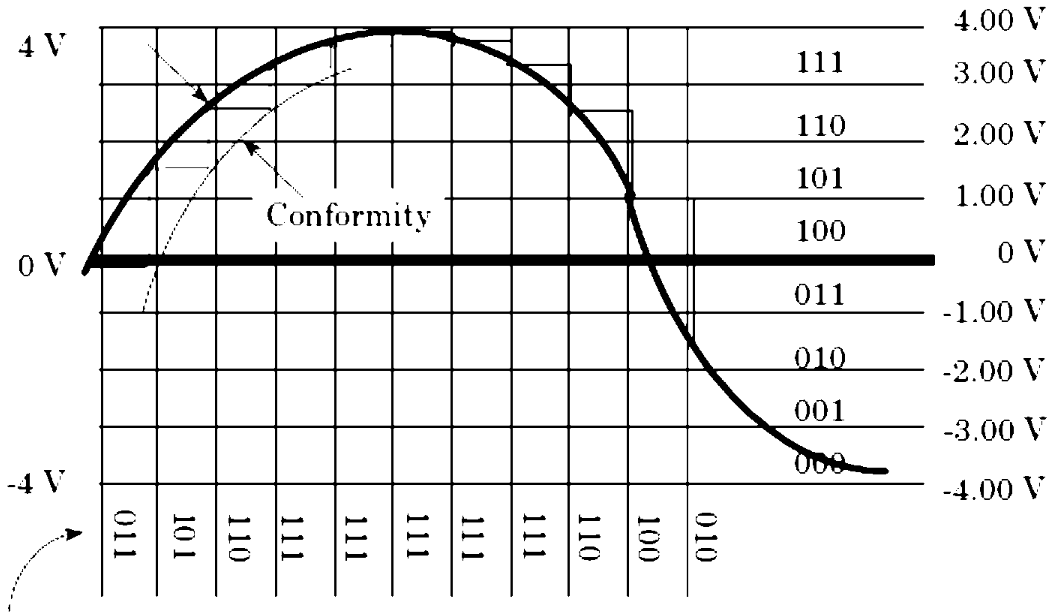
- Parallel communication speeds data
- Modules for input, output, & both
- I/O modules may also have microprocessor



Digital Conversion of Inputs and Outputs

- Signals, which would come in either from discrete or analog values, must now be converted to digital. Once the manipulation of these digital signals within the controller is complete, they must then be converted back to discrete and analog outputs.

One method to visualize analog to digital conversion



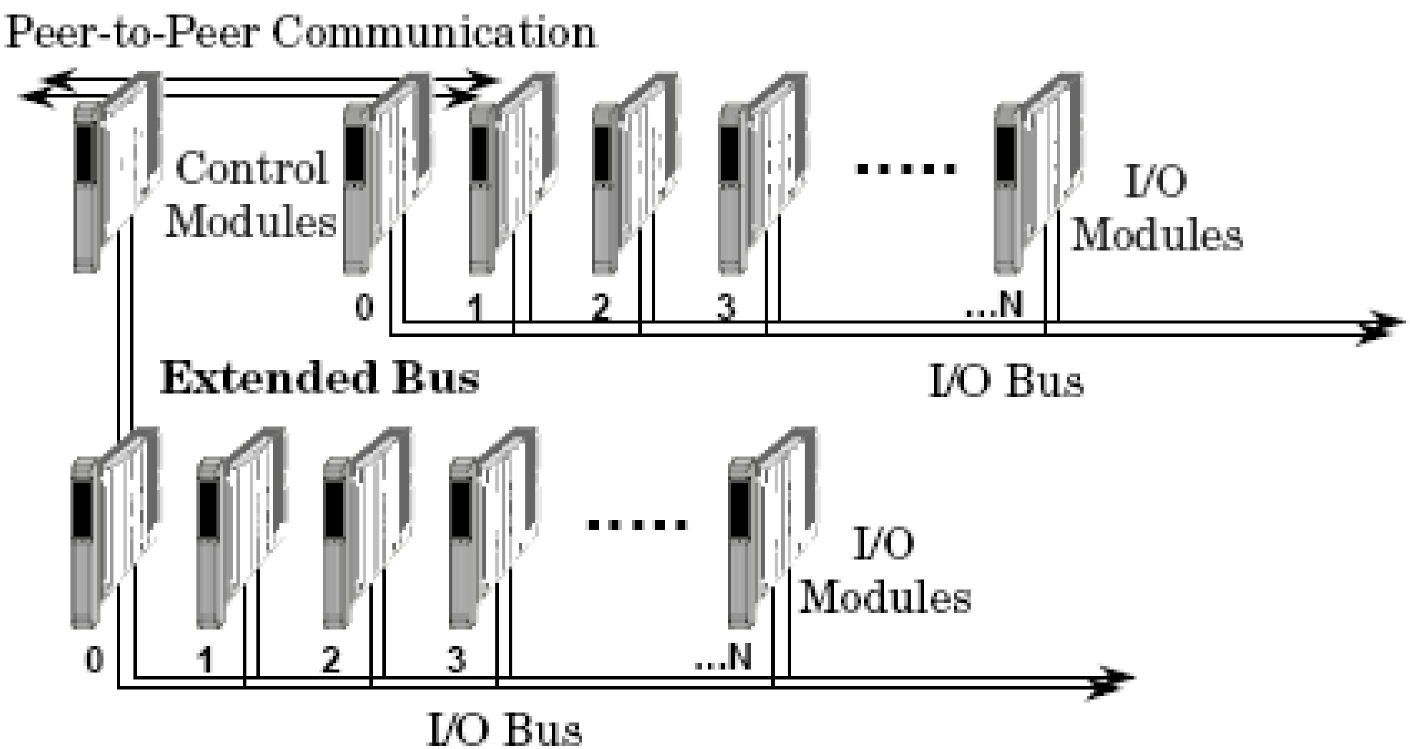
- 3 bit resolution conversion (a coarse conformity)
- 13 bit resolution ≈ 1 in 8000
- 14 bit resolution ≈ 1 in 16000

Remote Input/output Connections

- Remote I/O considered to be the controller’s own digital communication subnetwork, and quite often the connection is parallel rather than serial.

Remote Input/output Connections

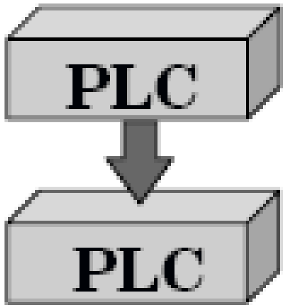
Remote I/O showing Peer-to-Peer Connections to Other Controller(s)



- When controller not needed in remote area
- Local & remote may be from same controller

Power Interruptions

- Where are you in "strategy cycle" upon switchover? ...Loops? ...Sequence state?
- Typically programming restarts at beginning...............................
- What about power outage?
  ...and return?



(Internal timer to mark duration,
...can act upon: Time/State/Conditions)

Power Interruptions

- While there are still controllers that must reload their configuration after a power interruption, many are equipped with built-in batteries to protect RAM memory, where the configuration resides.

- The charge on these batteries is often monitored, and significant warning time is provided through internal diagnostics.

- That warning, usually given weeks in advance, will display at the controller itself with an LED or alphanumeric code and will also be presented as part of the diagnostic video display(s) for the operator and others

Any control system needs …

- Inputs (analog and digital)
- Outputs (analog and digital)
  - Based on status of inputs
  - Various rules of behaviour
- Indication to an operator
  - Display of values
  - Alarms
- Operator input

Process plants

- Mostly analog inputs/outputs
- Mostly performing feedback control
- Needs complex arithmetic
- Changes take place slowly
- DCSs developed specially for this purpose

DCS – Typical I/O supported

Analog Input Modules
- High-Level Single-Ended
  - 1-5 Vdc
  - 4-20 mA
- High-Level Isolated
  - 0-10 Vdc
  - 4-20 mA
- High-Level Isolated
  - -10 to 70 mV
- Various RTD types
- Various Thermocouple types

Analog Output Modules
- 4 - 20 mA dc

Smart Modules
- HART Input
- HART Output

Discrete Input Modules
- 3 - 32 Vdc w/ debounce
- 3 - 32 Vdc w/o debounce (fast switching)
- 3 - 32 Vdc for Vortex flowmeters
- 90 - 140 Vac/Vdc
- 180 - 280 Vac/Vdc
- Dry Contact input (low side switching)

Discrete Output Modules
- 24 Vdc
- 3-60 Vdc
- 24-140 Vac
- 24-140 Vac w/ MOV-protection
- 24-280 Vac
- 24-280 Vac w/ MOV-protection
- Relay Output