Computer Networking

Software-Defined Networking (SDN)

Software-Defined Networking

- Motivation
- Enterprise network management
- Scalable SDN

- Readings:
  - A Clean Slate 4D Approach to Network Control and Management
  - Onix: A Distributed Control Platform for Large-scale Production Networks
- Optional reading
  - Ethane: Taking Control of the Enterprise

Software-Defined Networking

- Motivation
- Enterprise network management
- Scalable SDN

- Readings:
  - **A Clean Slate 4D Approach to Network Control and Management**
  - Onix: A Distributed Control Platform for Large-scale Production Networks
- Optional reading
  - Ethane: Taking Control of the Enterprise

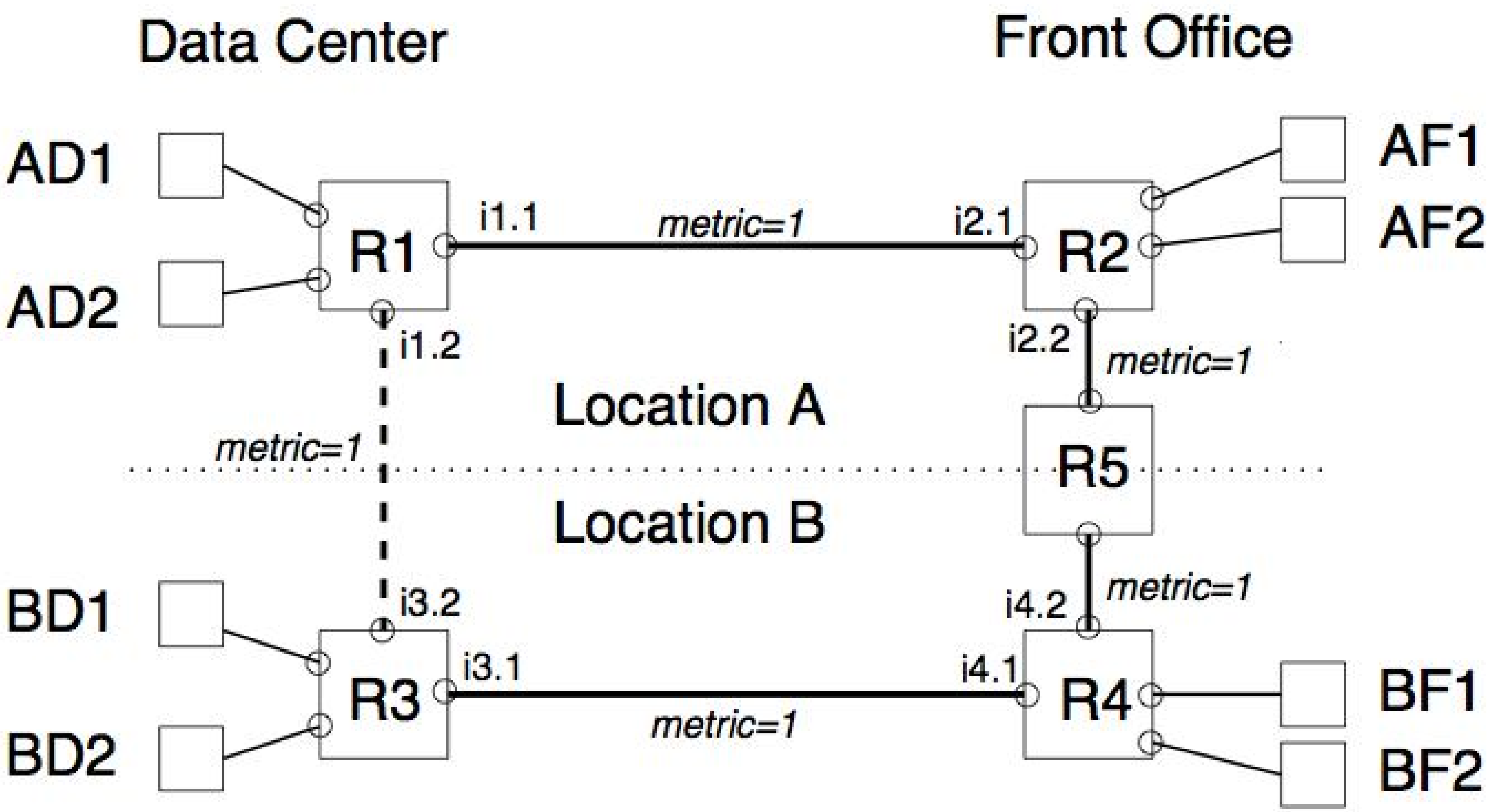
4D: Motivation

- Network management is difficult!
- Operators goals should be implemented as “workarounds”
- Observation: current Internet architecture bundles control logic and packet handling (e.g., OSPF)
- Challenge: how to systematically enforce various, increasingly complex high-level goals?

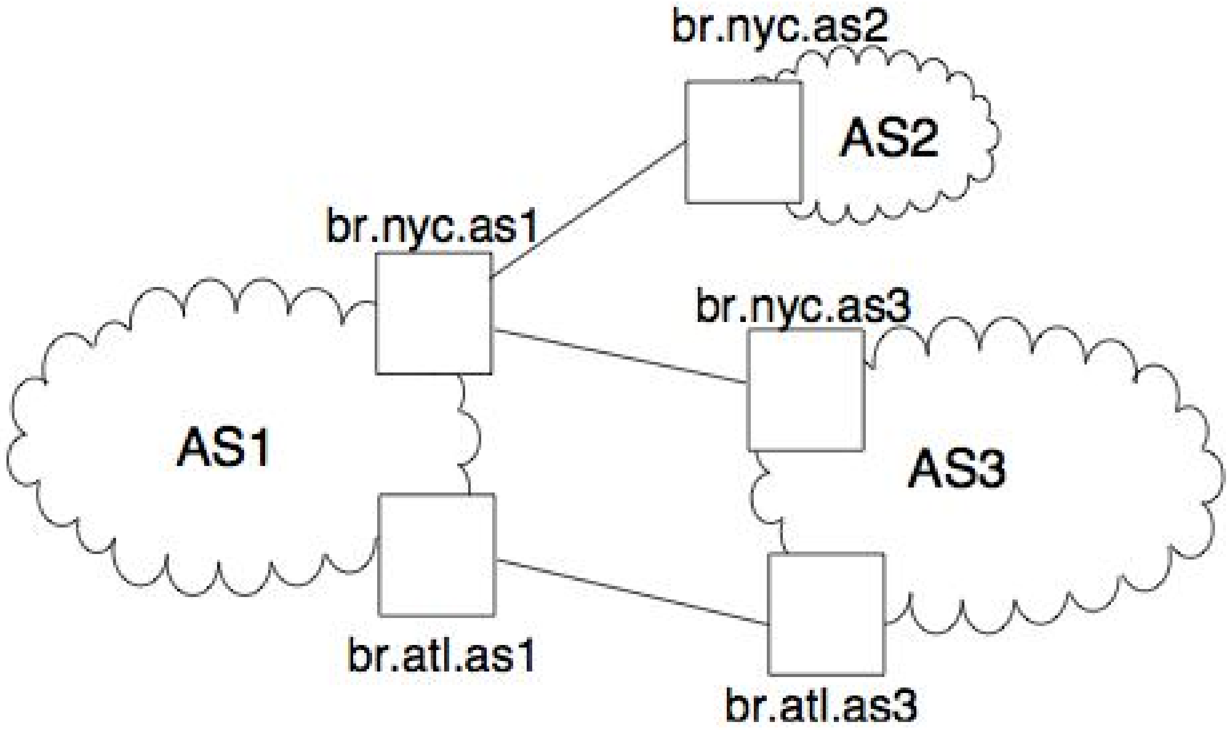
Design choices

- Incremental deployment
  - Advantage: easier to implement
  - Disadvantage: point solution?
- 4D advocates a clean-slate approach
  - Build control plane/network management from the ground up
  - Constraint: no change of packet formats
- Insight: Decouple the control and data planes

Example 1:
Front- Office Data Center ACL



Example 2: Spurious Routing



Management today

- Data plane
  - Packet forwarding mechanisms
- Control plane
  - Routing protocols
  - Distributed
- Management plane
  - Has to reverse engineer what the control plane
  - Work around rather than work with!

Driving principles

- Network-level objectives
  - High-level, not after-the-fact
- Network-wide views
  - Measurement/monitoring/diagnosis
- Direct control
  - No more “reverse engineering” or “inversion”
  - Direct configuration

4D Architecture

- Decision plane
  - routing, access control, load balancing, …
- Dissemination plane
  - control information through an independent channel from data
- Discovery plane
  - discover net. elements and create a logical net. map
- Data plane
  - handle individual packets given *state* by decision plane (e.g., forwarding tables, load balancing schemes,…)

Advantages of 4D Architecture

- Separate networking logic from distributed systems issues
- Higher robustness
- Better security
- Accommodating heterogeneity
- Enabling of innovation and network evolution

Challenges for 4D

- Complexity
- Stability failures
- Scalability problems
- Response time
- Security vulnerabilities

Research Agendas

- Decision plane
- Dissemination plane
- Discovery plane
- Data plane

Research Agendas

- Decision plane
- Dissemination plane
- Discovery plane
- Data plane

Research Agendas

- Decision plane
  - Algorithms Satisfying Network-Level Objectives
    - Traffic engineering
    - Reachability policies
    - Planned maintenance
    - Leveraging network structure
    - Multiple network-level objectives
    - Finding the right separation of timescales
  - Coordination Between Decision Elements
  - Introducing Hierarchy in the Decision Plane

Research Agendas

- Decision plane
  - Algorithms Satisfying Network-Level Objectives
  - Coordination Between Decision Elements
    - Distributed election algorithms
    - Independent DEs
  - Introducing Hierarchy in the Decision Plane

Research Agendas

- Decision plane
  - Algorithms Satisfying Network-Level Objectives
  - Coordination Between Decision Elements
  - Introducing Hierarchy in the Decision Plane
    - Large network managed by a single institution
    - Multiple networks managed by different institutions

Research Agendas

- Decision plane
- **Dissemination plane**
  - Connecting decision elements with routers/switches
  - Achieving direct control
- Discovery plane
- Data plane

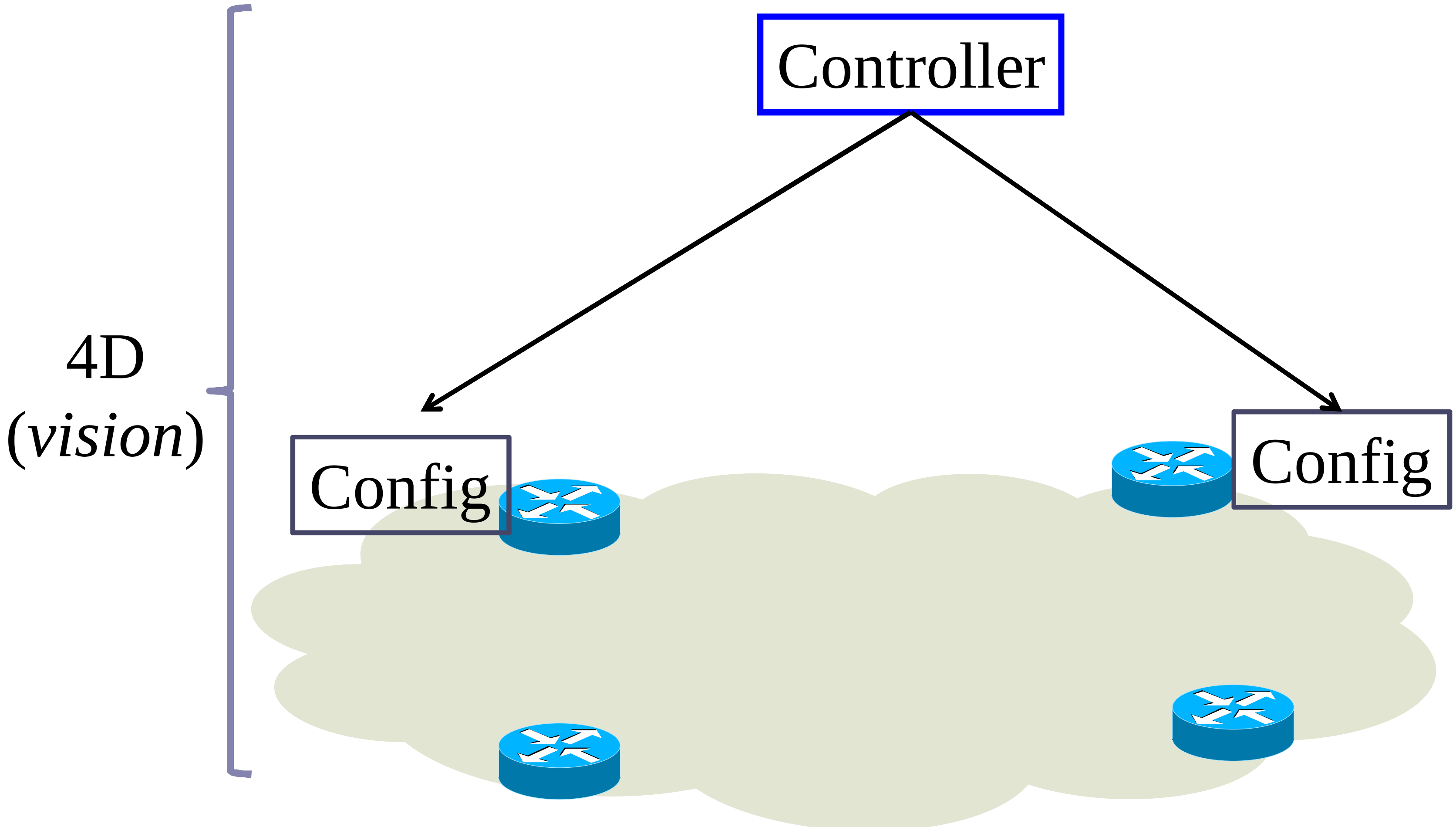
Research Agendas

- Decision plane
- Dissemination plane
- **Discovery plane**
  - Support for decision-plane algorithms
  - Bootstrapping with zero pre-configuration beyond a secure key
  - Supporting cross-layer auto-discovery
- Data plane

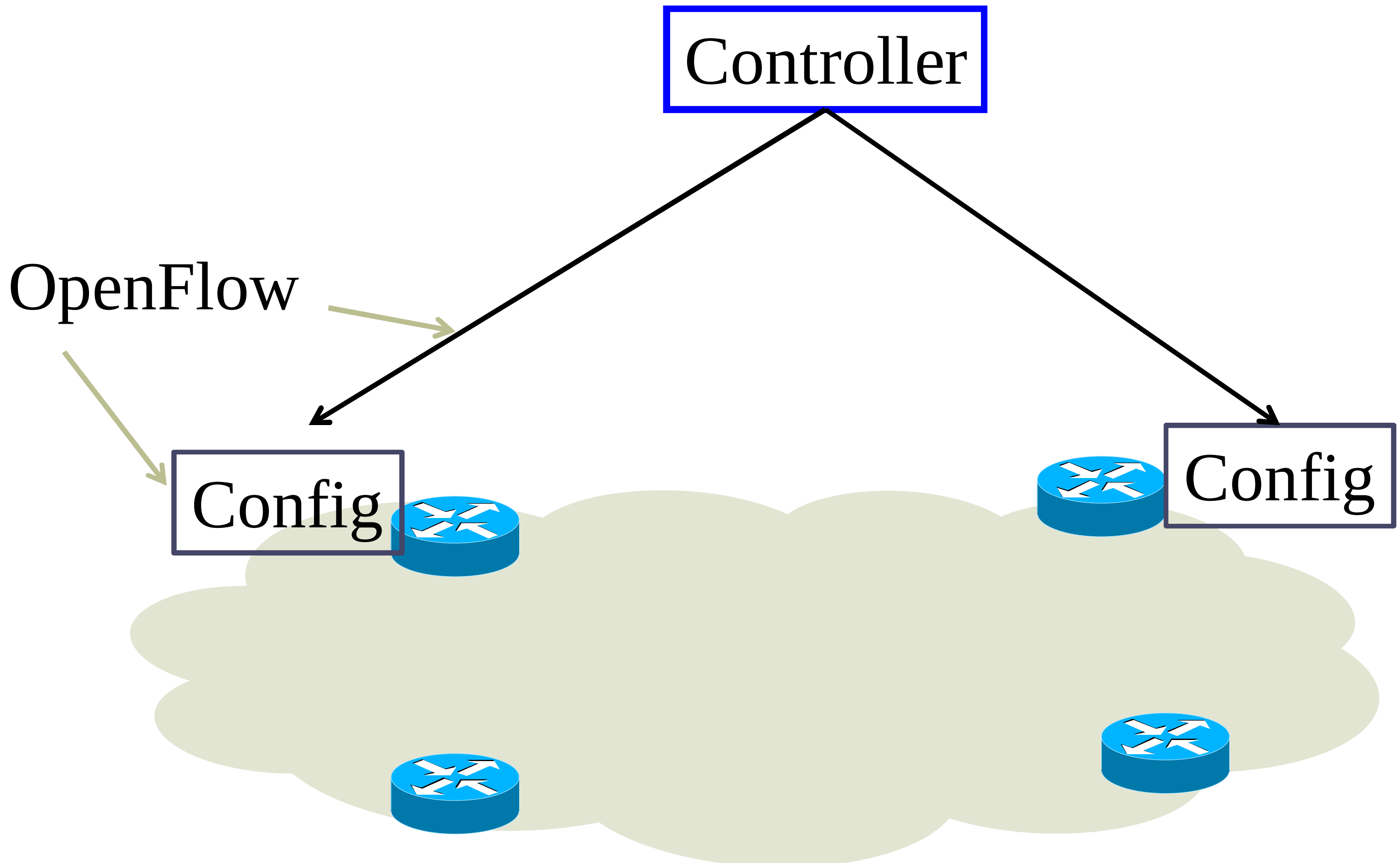
Research Agendas

- Decision plane
- Dissemination plane
- Discovery plane
- **Data plane**
  - Packet-forwarding paradigms
  - Advanced data-plane features

Where are we?

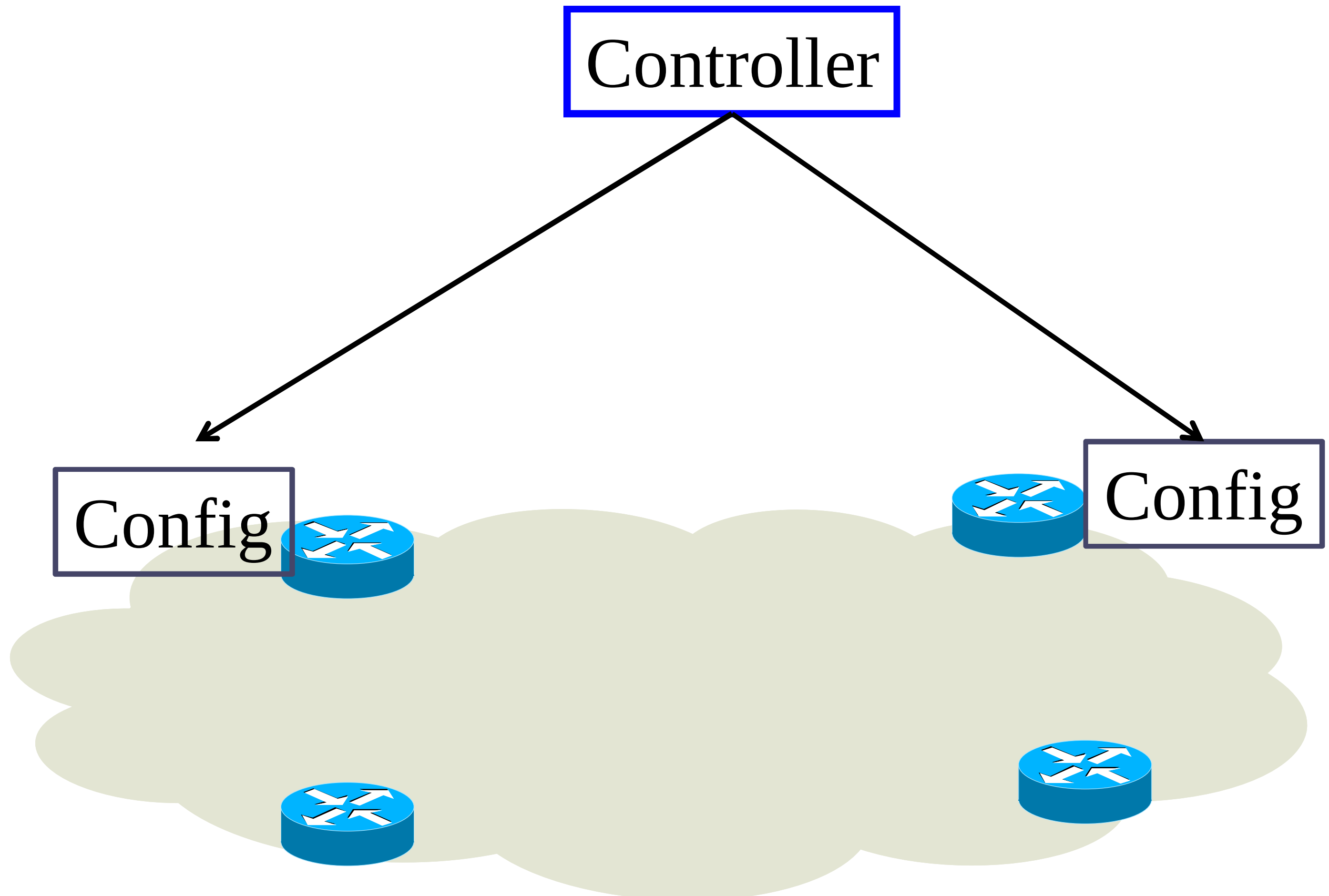


Where are we?

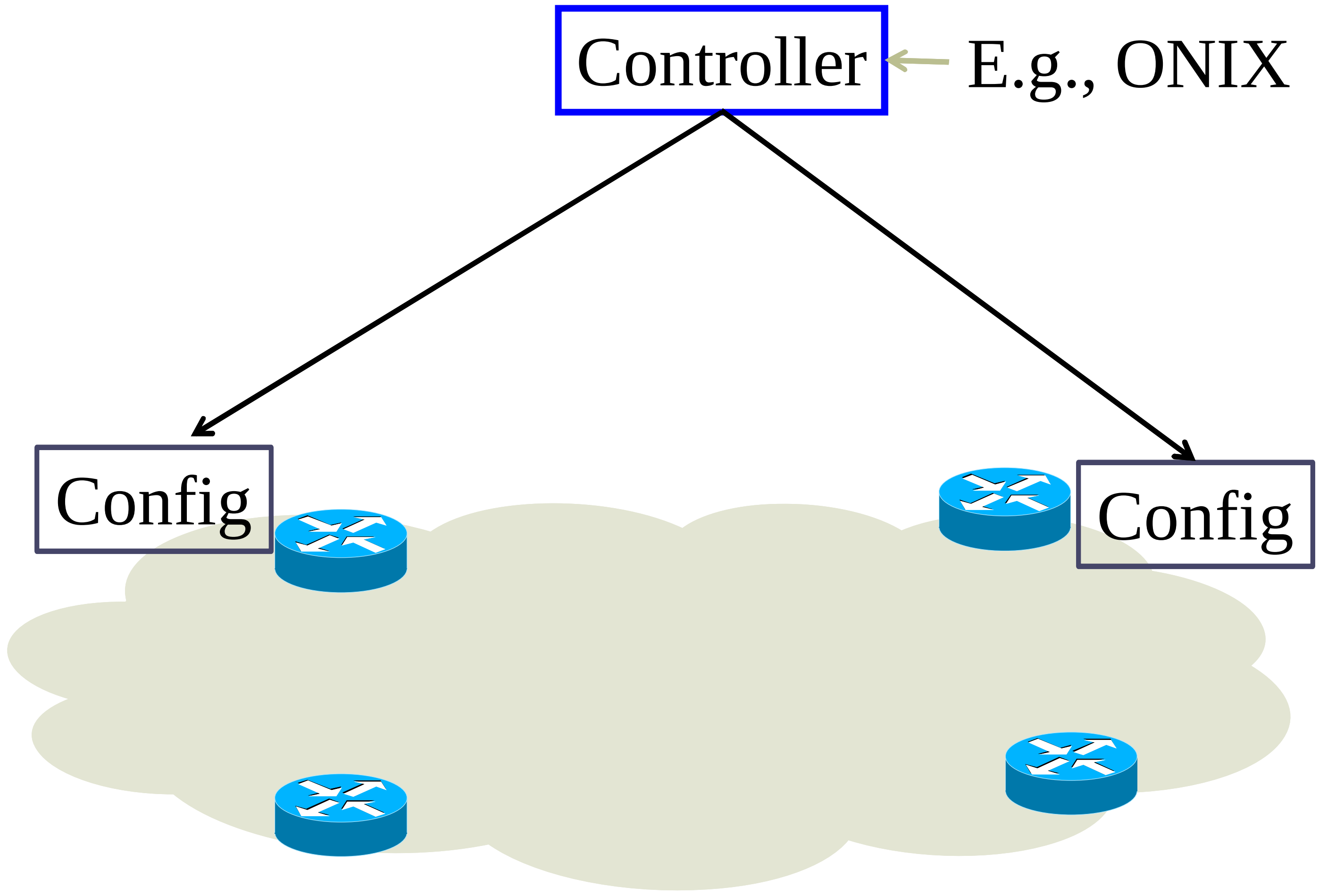


Where are we?

Ethane (*concrete example*)



Where are we?



Software-Defined Networking

- Motivation
- Enterprise network management
- Scalable SDN

- Readings:
  - A Clean Slate 4D Approach to Network Control and Management
  - Onix: A Distributed Control Platform for Large-scale Production Networks
- Optional reading
  - **Ethane: Taking Control of the Enterprise**

Motivation

- Enterprise configuration
  - Error prone: 60% of failures due to human error
  - Expensive: 80% of IT budget spent on maintenance and operations

- Existing solutions
  - Place middleboxes at chokepoints
  - Retrofit via Ethernet/IP mechanisms

Driving question

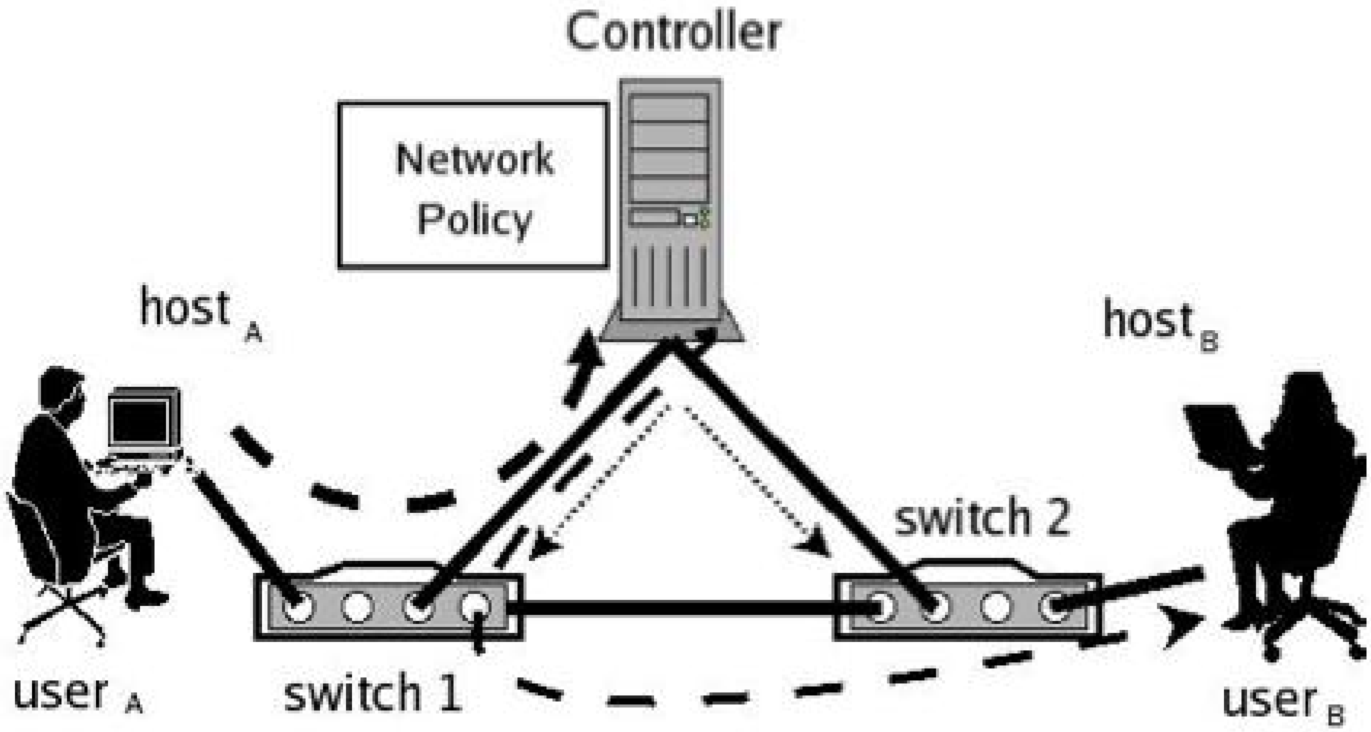
- Make enterprises more manageable

- What’s good about enterprises
  - Security policies are critical
  - Already somewhat centralized

Three principles in Ethane

- Descriptive/declarative policies
  - Tie it to names not locations/addresses
- Packet paths determined explicitly by policy
- Binding between packet and origin
  - No spoofing
  - Accountability

How Ethane Works

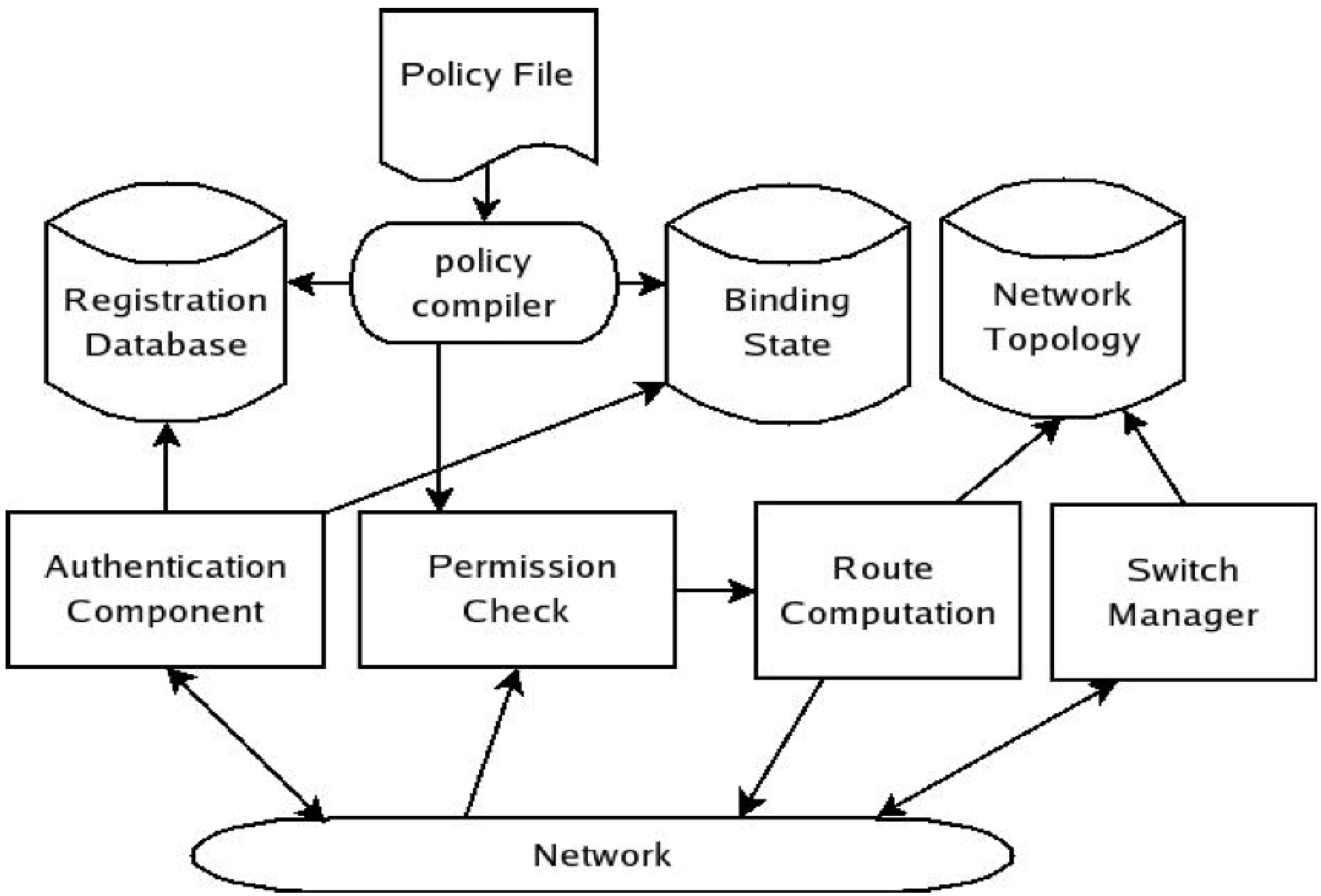


- First packet sent to Controller
- Subsequent packets use FlowTable
- No host-to-host communication without explicit permission

Ethane in use

1. Registration
   - explicit registration of users, hosts, and switches
2. Bootstrapping
   - spanning tree
3. Authentication
   - controller authenticates the host and assigns IP
   - user authenticates through a web form
4. Flow set up
5. Forwarding

Controller Design Components



- Explicit per-flow way-pointing

Switch Design

- Flow Table
- Local switch manager
- Secure channel to controller

Reliability

- Cold standby
  - Can potential lose some state

- Warm standby
  - Need some sort of consistency

- Fully replicated
  - Multiple active controllers

Policy Language

- Common tasks expressed as predicates
- Allow, deny, waypoint
- Interpret vs compile

Policy Language

```
# Groups —
desktops = ["griffin","roo"];
laptops = ["glaptop","rlaptop"];
phones = ["gphone","rphone"];
server = ["http_server","nfs_server"];
private = ["desktops","laptops"];
computers = ["private","server"];
students = ["bob","bill","pete"];
profs = ["plum"];
group = ["students","profs"];
waps = ["wap1","wap2"];
%%
# Rules —
[(hsrc=in("server")∧(hdst=in("private"))] : deny;
# Do not allow phones and private computers to communicate
[(hsrc=in("phones")∧(hdst=in("computers"))] : deny;
[(hsrc=in("computers")∧(hdst=in("phones"))] : deny;
# NAT-like protection for laptops
[(hsrc=in("laptops")] : outbound-only;
# No restrictions on desktops communicating with each other
[(hsrc=in("desktops")∧(hdst=in("desktops"))] : allow;
# For wireless, non-group members can use http through
# a proxy. Group members have unrestricted access.
[(apsrc=in("waps"))∧(user=in("group"))] :allow;
[(apsrc=in("waps"))∧(protocol="http)] : waypoints("http-proxy");
[(apsrc=in("waps"))] : deny;
[]: allow; # Default-on: by default allow flows
```

Potential resource concerns

- Controller “DDoS”

- Controller scalability

Evaluation

- Mostly “feasibility”
- Trace-driven evaluations
- Failure emulation
- Scalability of request rate
- End-to-end performance

Ethane Prototype

- 300 hosts in CS department at Stanford
- Multiple “switches”
  - Wireless access point, linux, netfpga
- Controller
  - Standard linux PC
  - Linux PC (1.6GHz Celeron CPU and 512MB of DRAM)
- Controller handles 10,000 flows per second

Experiences

- Once deployed, easy to manage
- Add new switches, users is easy
- Journaling helps debugging
- Adding new features is easy

Advantages of Ethane

- Switches
  - Dumb
  - No complex distributed protocol
  - Focus purely on forwarding
  - Save forwarding rule space (try to keep only “active” flows)

Comments on Design

- Common vs worst case design?

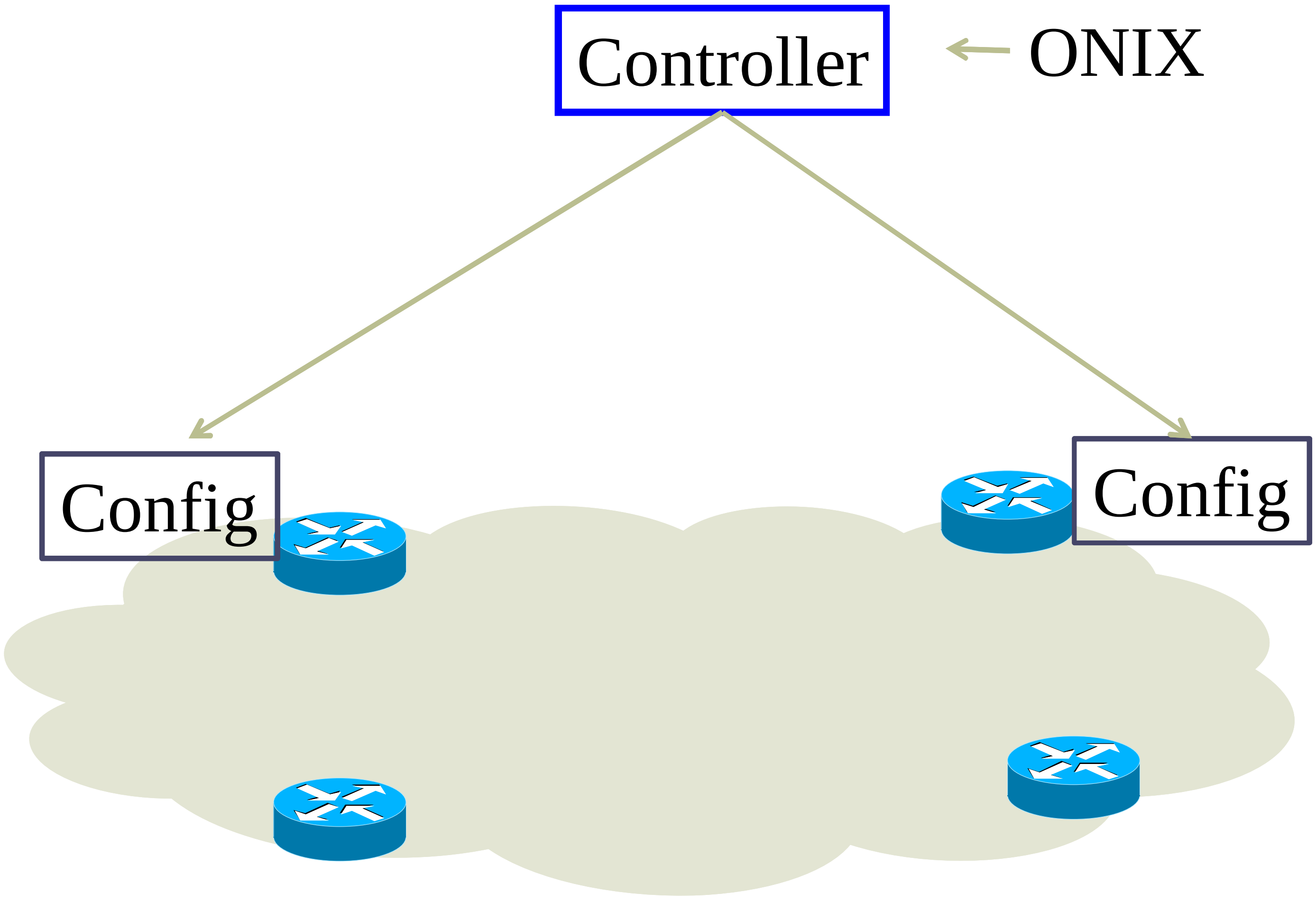
- Latency, scalability

Software-Defined Networking

- Motivation
- Enterprise network management
- Scalable SDN

- Readings:
  - A Clean Slate 4D Approach to Network Control and Management
  - **Onix: A Distributed Control Platform for Large-scale Production Networks**
- Optional reading
  - Ethane: Taking Control of the Enterprise

ONIX

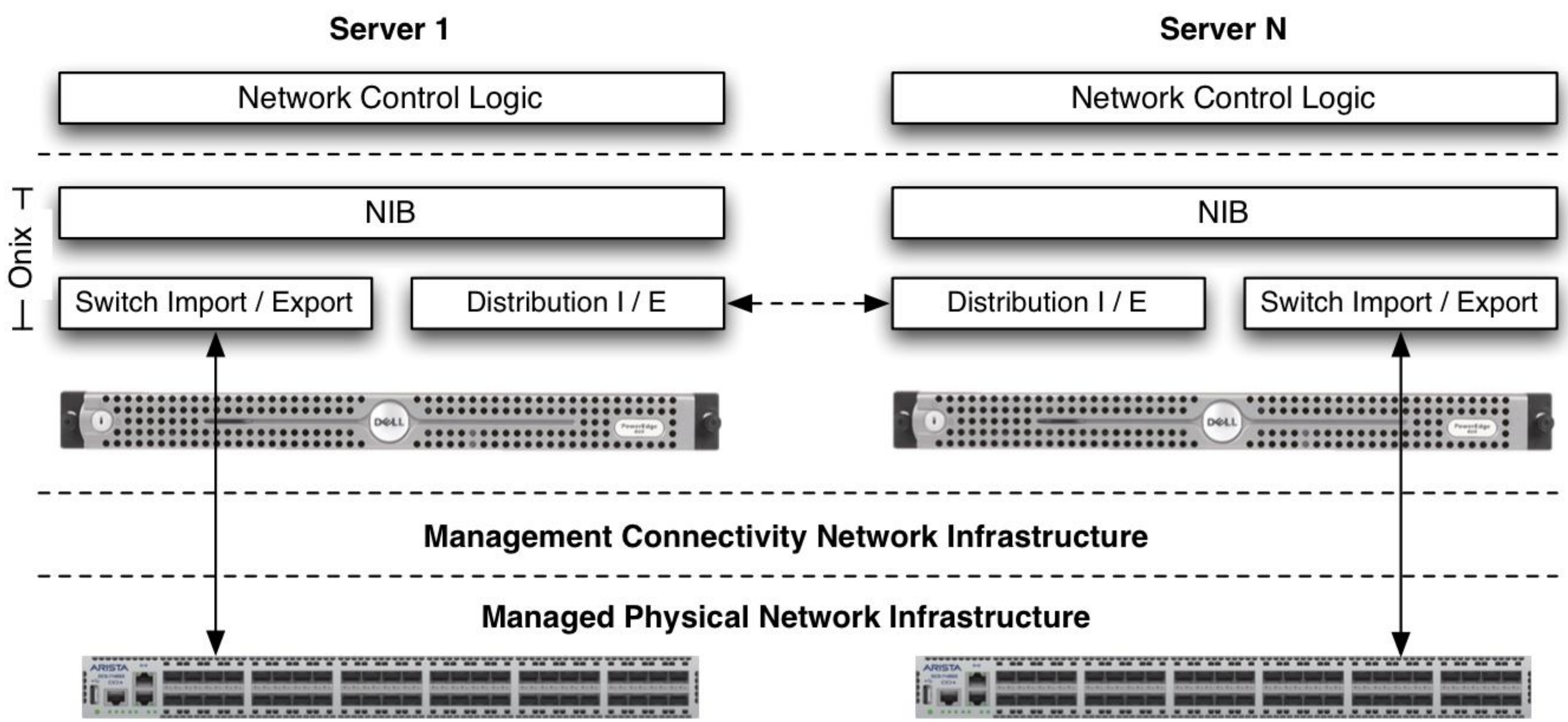


ONIX: How to build a controller platform?

What are the key challenges?

- Usability
- Performance
- Flexibility
- Scalability
- Reliability/availability

ONIX



Server 1
Server N
Network Control Logic
Network Control Logic
Onix
NIB
NIB
Switch Import / Export
Distribution I / E
Distribution I / E
Switch Import / Export
Management Connectivity Network Infrastructure
Managed Physical Network Infrastructure

ONIX Design Decisions

- “Data-centric” API

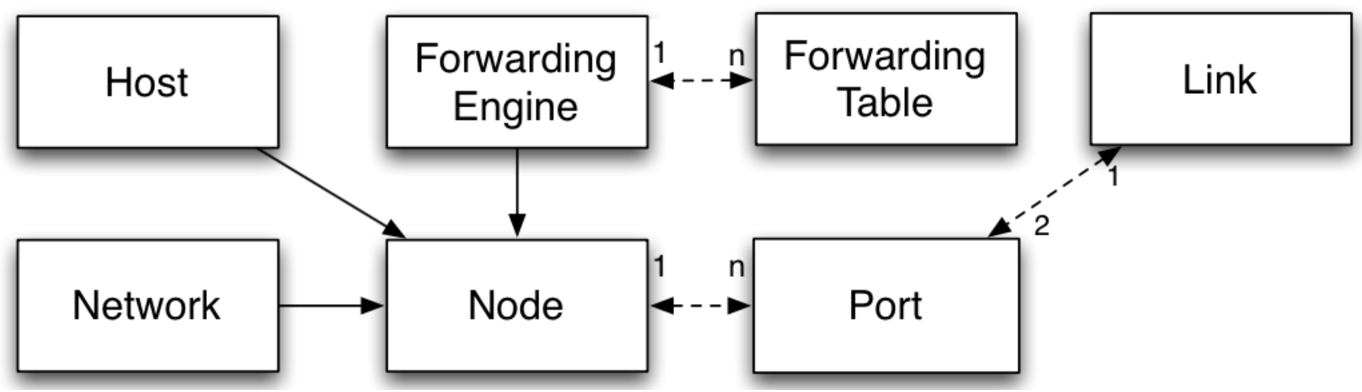
- Treat all networking actions as data actions
  - Read
  - Alter
  - Register for changes in network state

Core component == NIB

- Network information base
  - Analogous to forwarding information base
- Graph of all network entities
  - Switches, ports, interfaces, links etc
- Applications read/register/manipulate NIB

Core component == NIB

- NIB is a collection network entities
- Each entity is a key-value pair



Default network entity classes

ONIX NIB APIs

Category	Purpose
Query	Find entities.
Create, destroy	Create and remove entities.
Access attributes	Inspect and modify entities.
Notifications	Receive updates about changes.
Synchronize	Wait for updates being exported to network elements and controllers.
Configuration	Configure how state is imported to and exported from the NIB.
Pull	Ask for entities to be imported on-demand.

Functions provided by the ONIX NIB API

Three scalability strategies

- Partition
  - Can we split the state into independent sub-sets?
  - E.g., different subnet forwarding rules on a switch

- Aggregate
  - zoom-in/zoom-out at different aggregation levels

- Tradeoff with weaker consistency/durability
  - E.g., replicated transactional DB for network topology
  - E.g., one-hop DHT for link utilization info

Two types of datastores

- DHT with weak eventual consistency
  - Used for “high” churn events
  - Frequent updates

- Transactional store with strong guarantees
  - Used for “low” churn events
  - E.g., network policy

Reliability

- Network element failure
  - discovered by traditional data plane mechanisms
  - application is in charge of deciding about the alternative policy after node/link failure
- ONIX instance failure
  - Option 1: other instances detect failure and take over
  - Option 2: have multiple instances manage a network element the network at all times
- Infrastructure failure
  - Use dedicated control backbone

Killer apps for ONIX

- Why did VMWare bought Nicira maybe?

- DVS

- Multi-tenant virtualization

Lingering questions

- flexibility
- Performance bottlenecks
- Consistency/conflicts

Summary

- 4D: An extreme design point
- Ethane: End-to-end enterprise network management
- ONIX: A distributed control platform

Next Lecture

- Network verification
- Readings:
  - HSA: Read in full
  - NOD: Read intro
  - Veriflow: Optional reading

UNIT- II

PLC AND ITS APPLICATION

Introduction

- **PLC(Programmable Logic Controller)**
- **Advantages of PLC**
- **Programming--- *Ladder diagram***
- **PLC in batch processing & others**
- **Current trends in the industry**

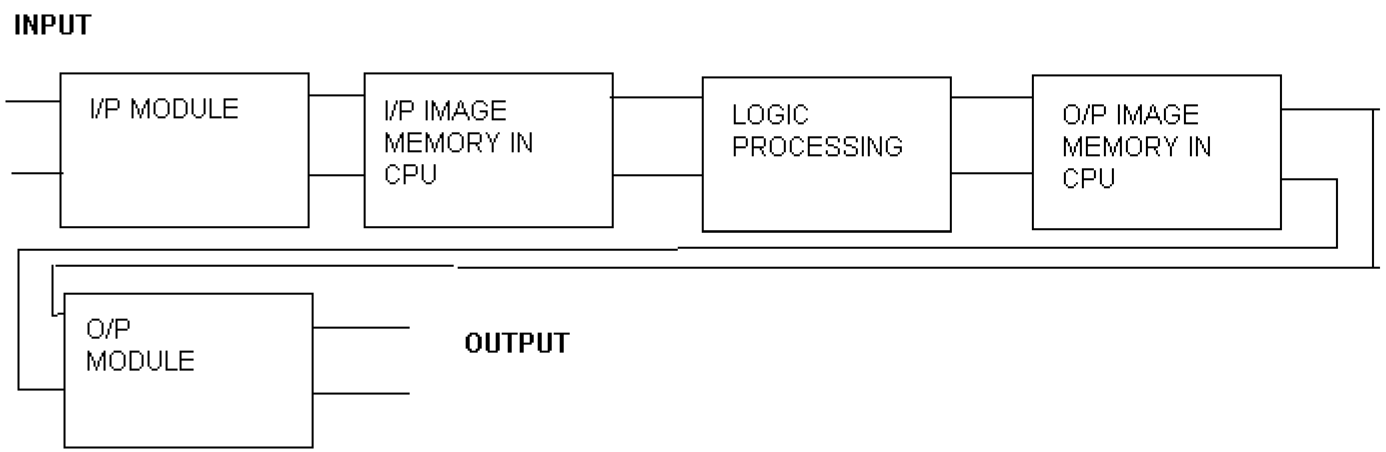
PLC(PROGRAMMABLE LOGIC CONTROLLERS)

1. **Replaces relays as logic elements.**
2. **Software oriented.**
3. **Fixed number of input-outputs/PLC .**
4. **Programmable.**
5. **Each I/O can be used as many times as necessary.**
6. **Downloaded from PC by a cable by the programming port .**

SCAN PROCESS OF A PLC

Steps in PLC scan process

- **I/P processing**
- **Program processing**
- **O/p processing**



Advantages Of PLC:

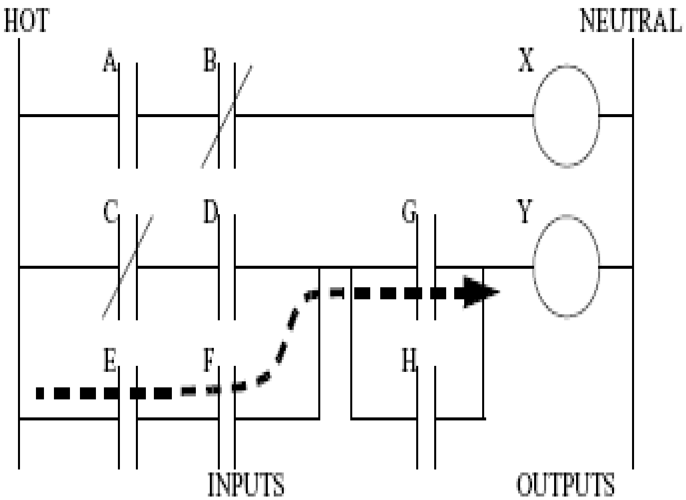
1. Flexible
2. Cost effective .
3. Greater Computational abilities.
4. Trouble shooting easier.
5. Reduce downtime.
6. Reliable.

PLC Programming

Ladder programming is mostly used

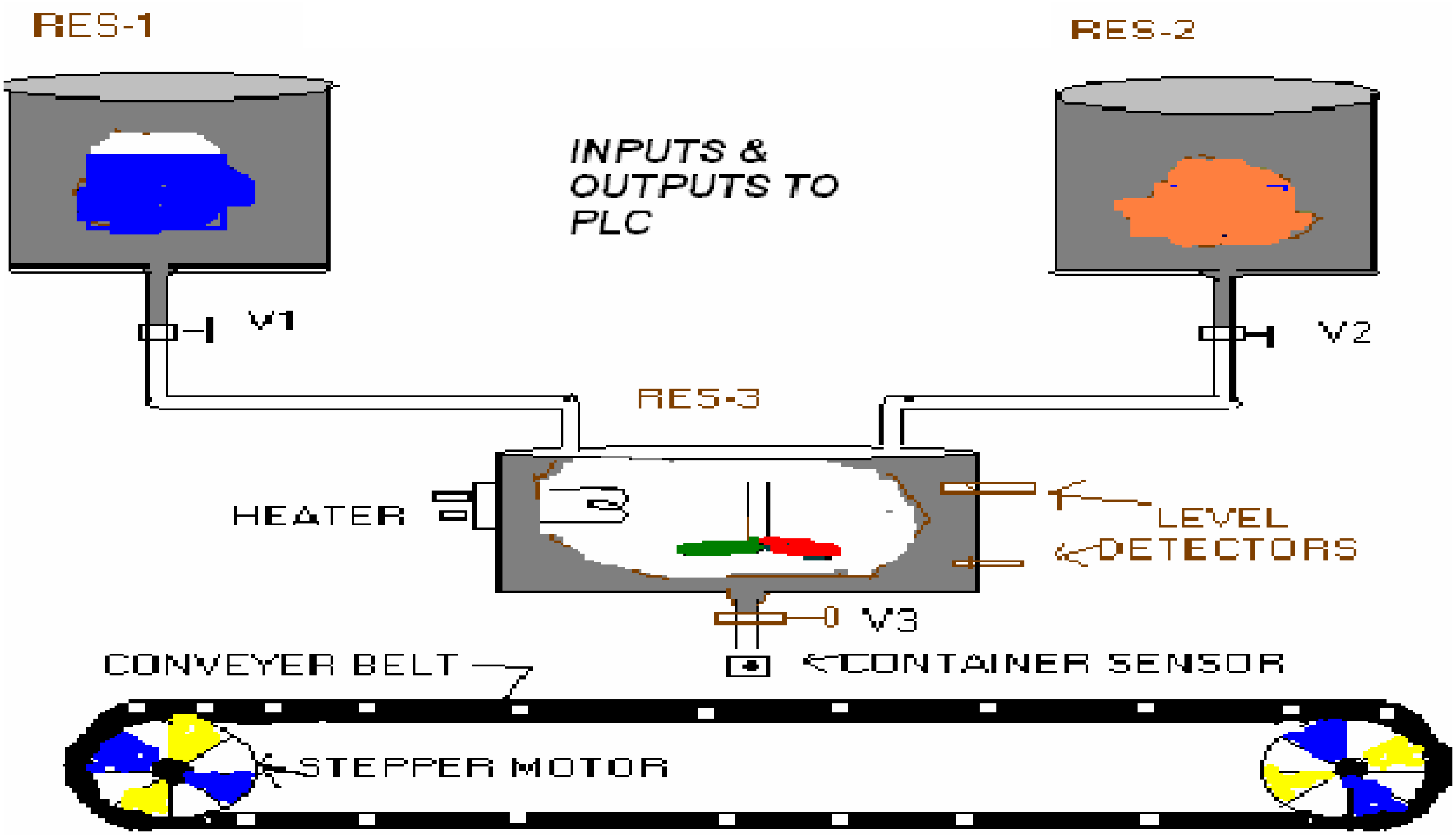
Ladder Programming has

- •Hot rail
- •Neutral line
- •Rungs
- •Input
- •Output

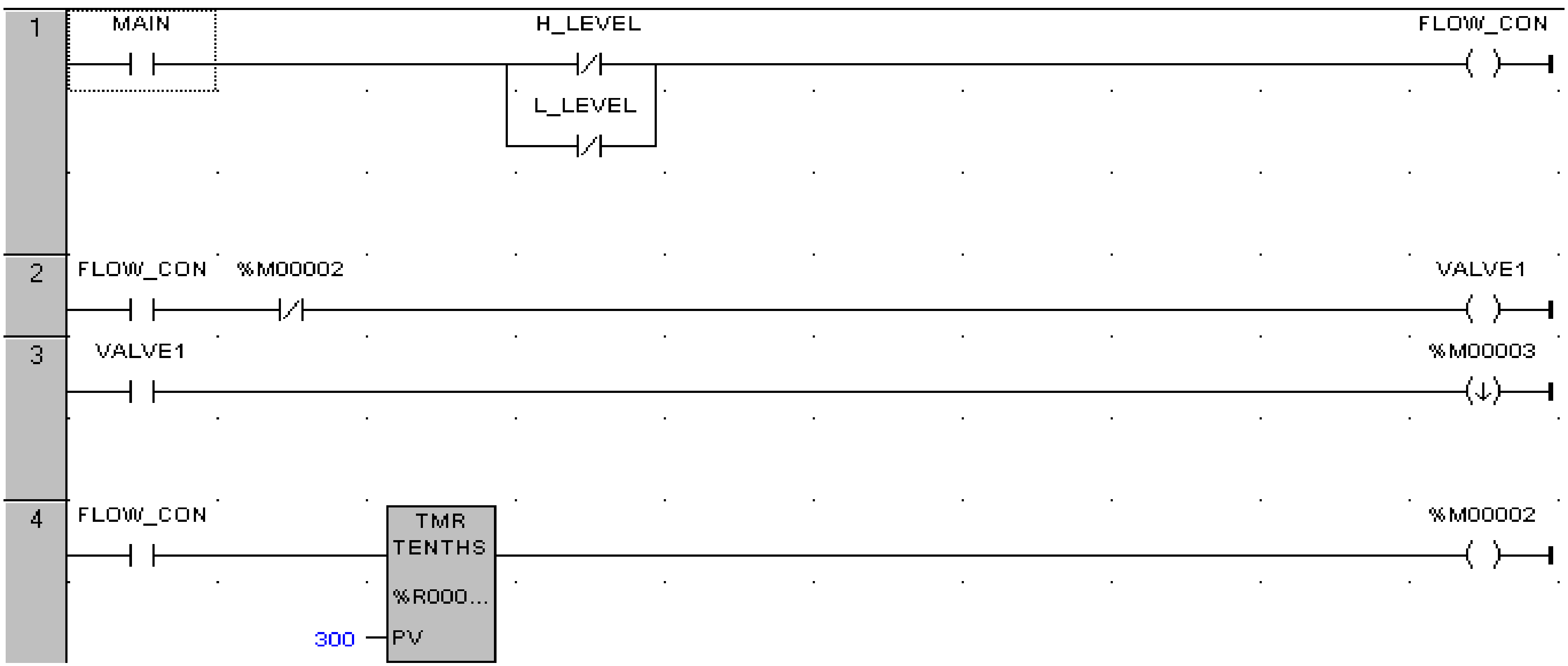


Note: Power needs to flow through some combination of the inputs (A,B,C,D,E,F,G,H) to turn on outputs (X,Y).

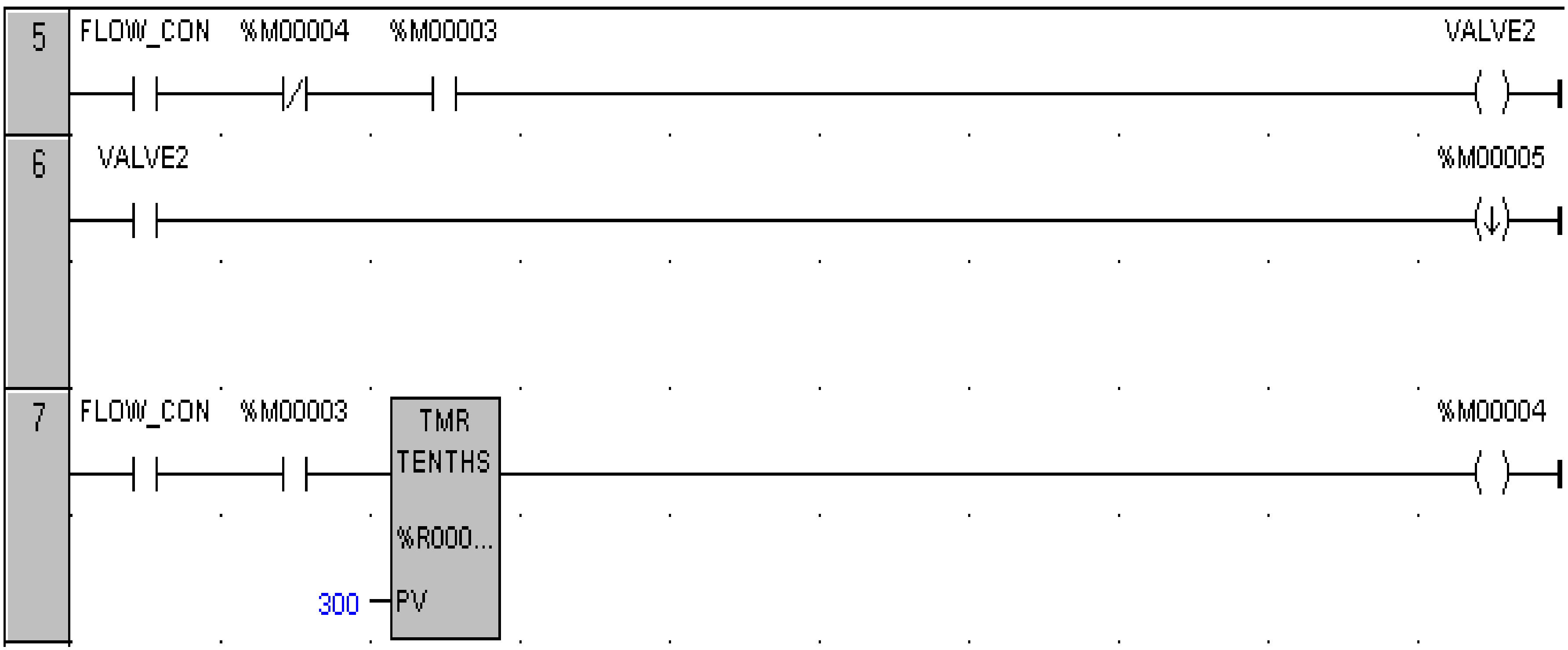
Batch Processing



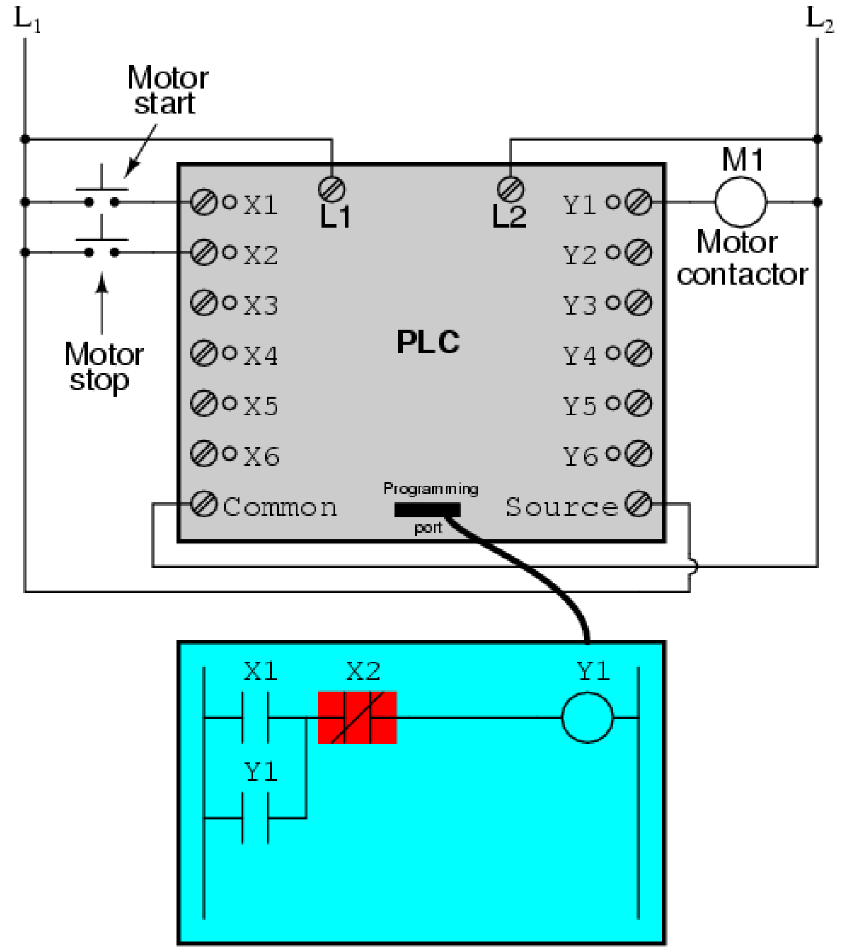
Ladder Diagram For Batch Processing (VALVE1)



Ladder Diagram (VALVE2)



Speed Control Of A Motor



Applications Of PLC

1. Petrochemical industry --crude separation etc..
2. Steel industry-- smelter operations, blast furnace, temperature monitoring etc.
3. Power generation -- boiler control from water injection, temperature, fuel, steam flow monitoring etc.
4. Process industries --air flow control, controlling air-fuel ratios etc.
5. Chemical industries-- proportion of chemicals .
6. Nuclear power generation plants .
7. Home automation .
8. PLCs in all phases of automated industrializations.

Different PLCs

GEFANUC

Inputs—14
Outputs—10
Software used—Versapro (Windows gui)
Power supply—24v

MESSUNG

Inputs—8
Outputs—6
Software used—Doxmini (dos)
Power supply—24v

Current Trends in the Industry

- “Smart" PLCs microprocessors memory.
- Multitasking.
- Multiaxis control with sophisticated vision systems .
- Hardened programmers Online & offline software development
- Menu-driven software and concurrent operating systems.

Conclusion

- **Hence we see that PLC are widely used in industries.**
- **PLCs have been gaining popularity on the factory floor and will remain predominant for some time to come.**
- **Future of advanced automation.**
- **Better control and management achieved.**

UNIT III

COMPUTER CONTROLLED SYSTEMS

Computer Control System

Expand present control system:

- DEC Alpha (VMS) + switched Ethernet
- CAMAC and VME crates + μVAX controllers
- X-terminal and PC consoles
- EPICS applications and GUI tools
- EPICS Channel Access to existing control system
- Database (Oracle RDB)

New interface hardware and development:

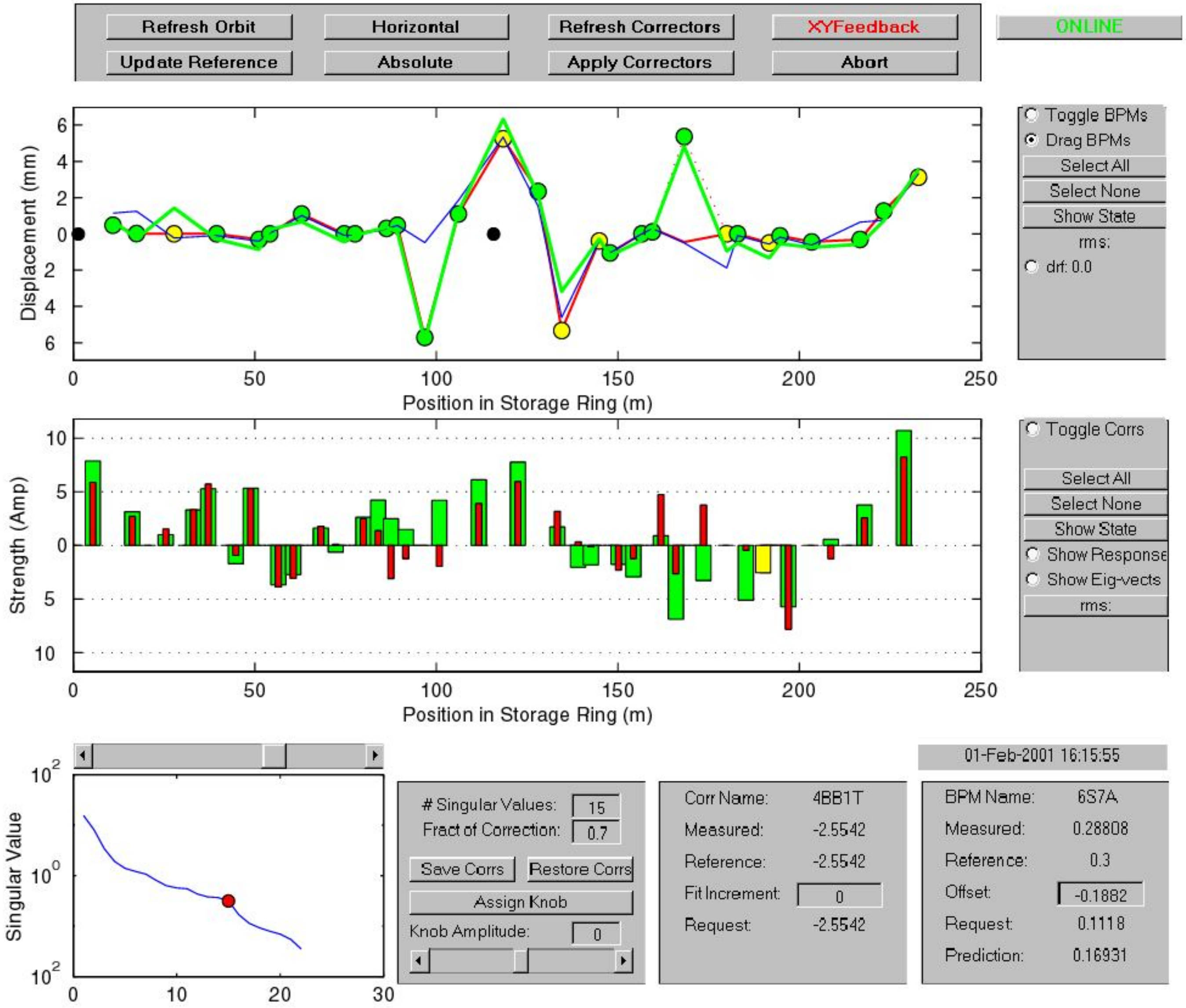
- Main power supply controllers (slow)
  Bitbus control, obsolete microcontrollers
- Fast power supply digital controller - develop
  Fast Ethernet (100 Mb/s) + switch
- BPM Processor, Orbit Feedback interface
  Power PCs
- RF Control System
  EPICS IOC (NI 68030 or replacement PPC), VXI crates

Software development:

- Intelligent crate controllers
  local process and control, data logging
- Power Supply controllers , BPM Processor, Orbit Feedback
  drivers, control programs, DSP code
- RF Control System
  EPICS,unix development system
  VxWorks, Matlab
- Application software

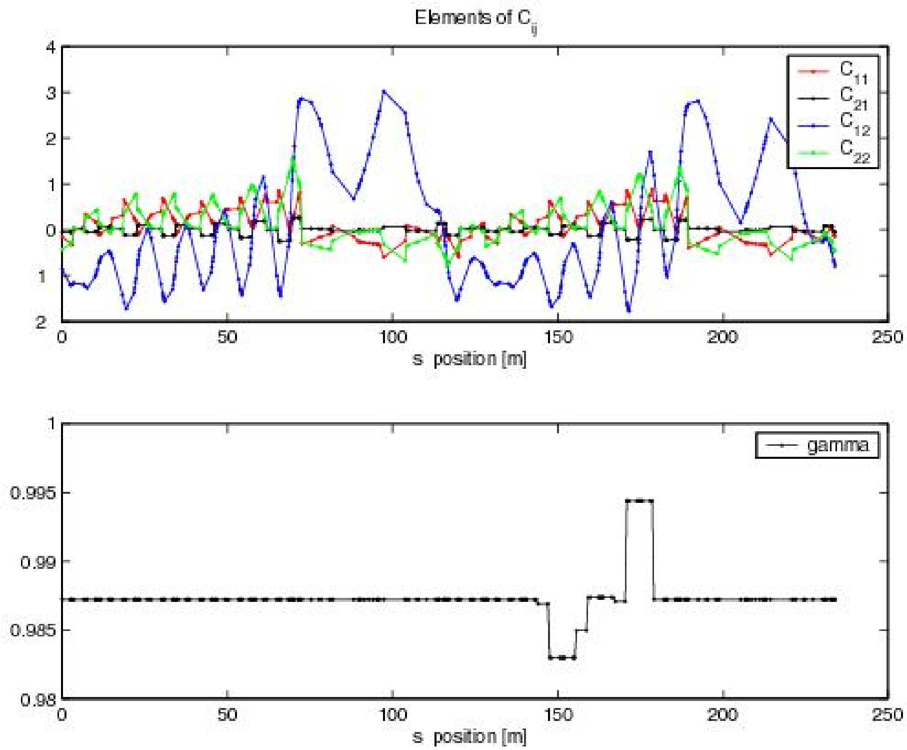
  VMS, EPICS, Matlab

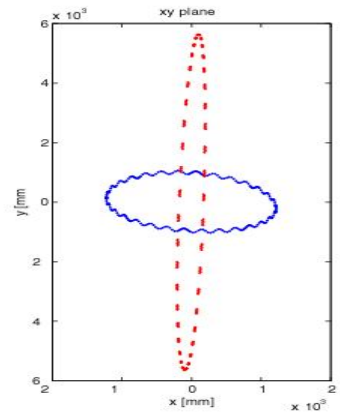
Orbit Control with Matlab and EPICS Channel Access

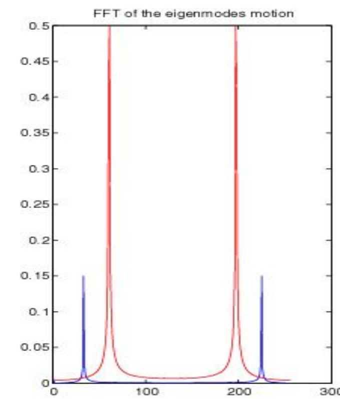


Matlab-based Accelerator Toolbox and Simulator

H-V coupling error







VME Crates and CPUs

VGM5 VME Dual PPCG4/G3 CPU Board (Synergy)

Dual or single CPUs in a single slot solution

Advanced PowerPC G4/G3 architecture

300-466 MHz CPU speed

Backside L2 cache 1 or 2 MB per CPU

PØ-PCI(TM) secondary data bus, ~264 MB/s

16-512 MB high-speed SDRAM

Up to 9 MB Flash

Supports industry-standard PMC I/O

Autosensing 10/100Base-TX Ethernet

Two serial ports standard; SCSI option

4-digit clock/calendar chip is Y2K compliant

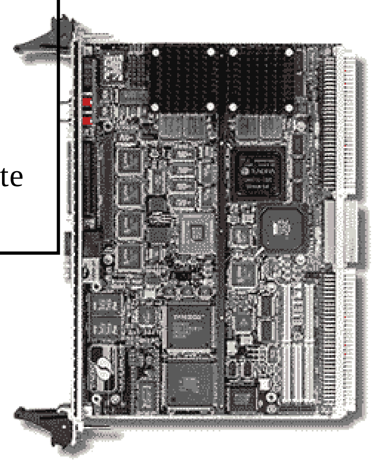
Supports VxWorks, Linux

Supports RACEway with PXB2 PMC module

VME64x support

VME Speedway doubles non-block transfer rate

Conformal coating option



VME Crates (Wiener)

21 slots, 6U VME cards

3U space for fan tray and plenum chamber

Card guides and ejector rails IEEE 1101.10

Monolithic backplane VME64x or VIPA

Microprocessor controlled fan-tray unit UEL 6020 with high efficient DC-fans (3 ea.), alphanumeric display, variable speed fan

Temperature control, front or bottom air inlet

Up to 8 temperature sensors in bin area with network option for remote monitoring and control (CAN-bus)

Remote CPU reset capability

Used at SLAC, BNL, CERN, BESSY, etc.

Beam Monitoring and Feedback Systems

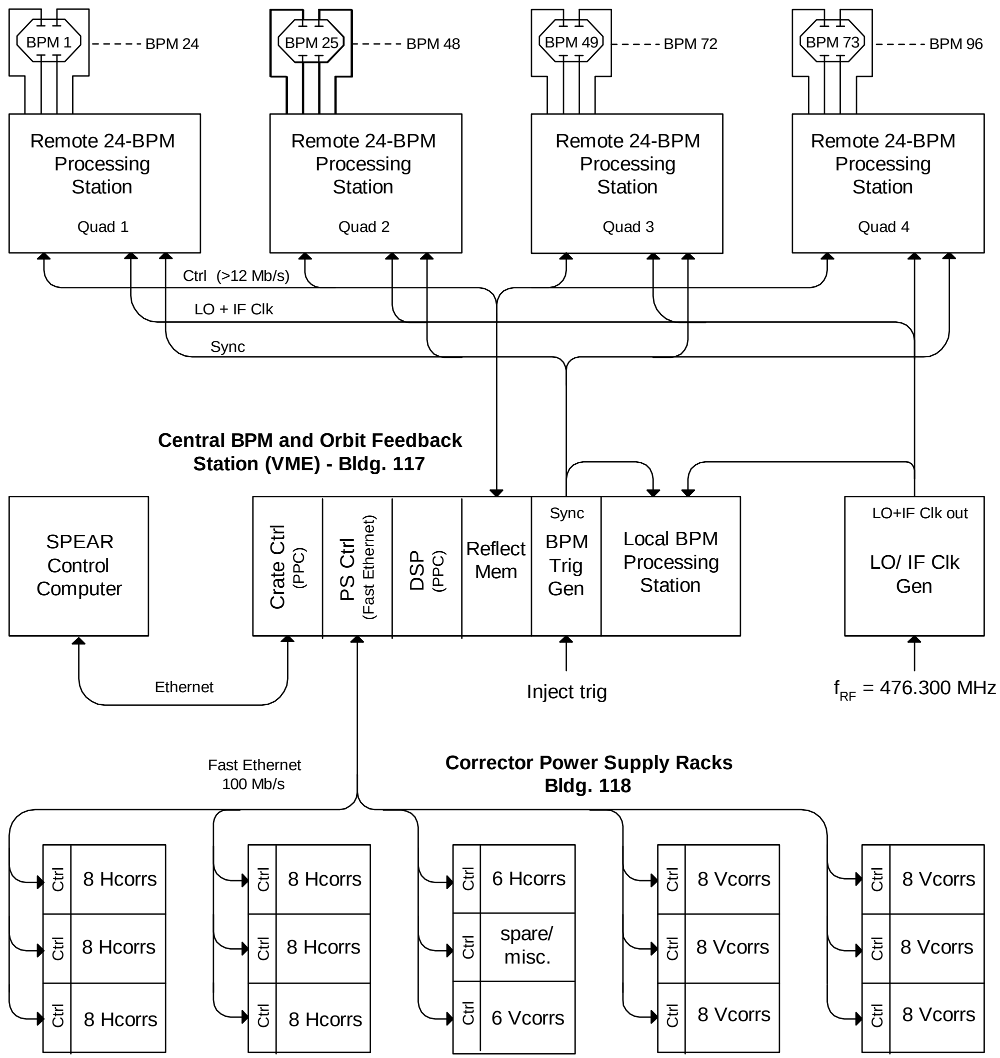
New for SPEAR 3:

- BPM Processing System
- Orbit Feedback System
- DCCT
- Scraper Controls
- Tune Monitor
- Synchrotron Light Monitor
- Quadrupole Modulation System

From SPEAR 2:

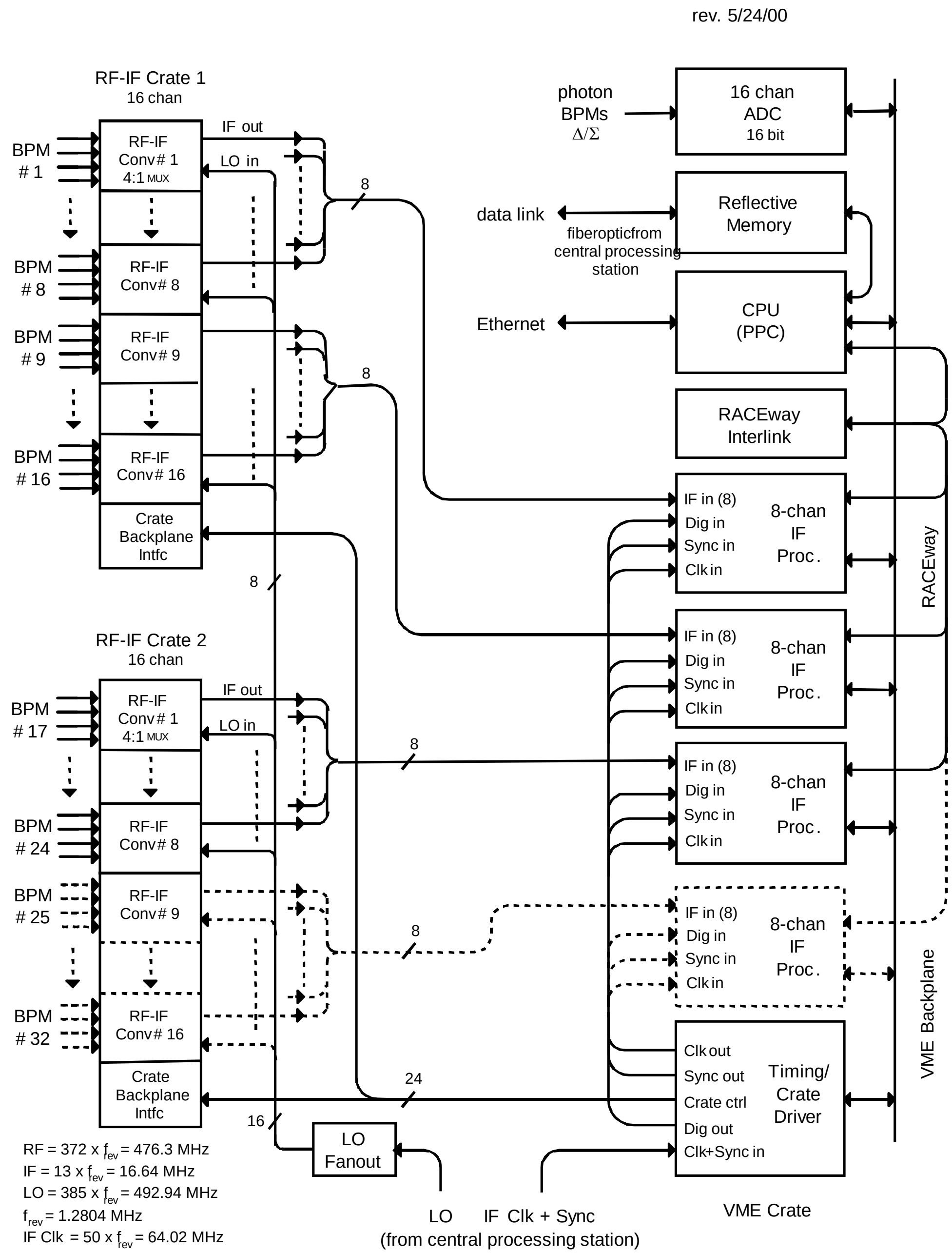
- Upgraded injection monitors
- Longitudinal Bunch Phase Monitor
- Transverse Bunch Phase Monitor

BPM Processing and Orbit Feedback System

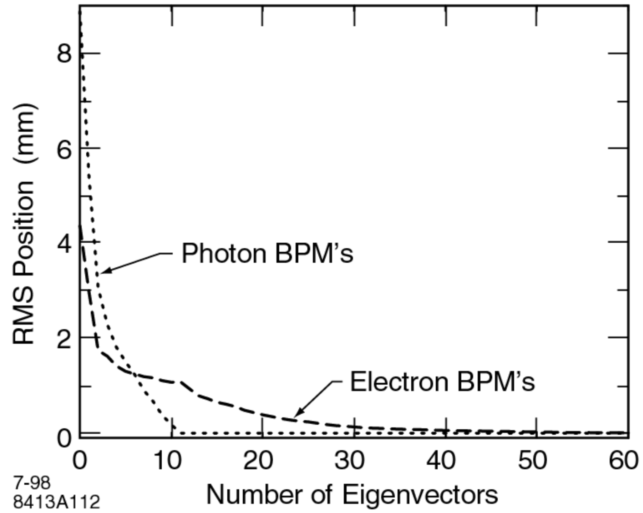


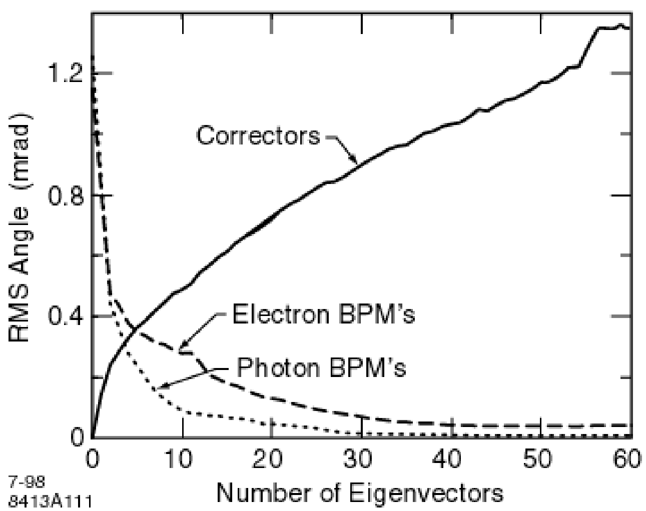
SPEAR 3 BPM and Orbit Feedback System rev. 5/24/00

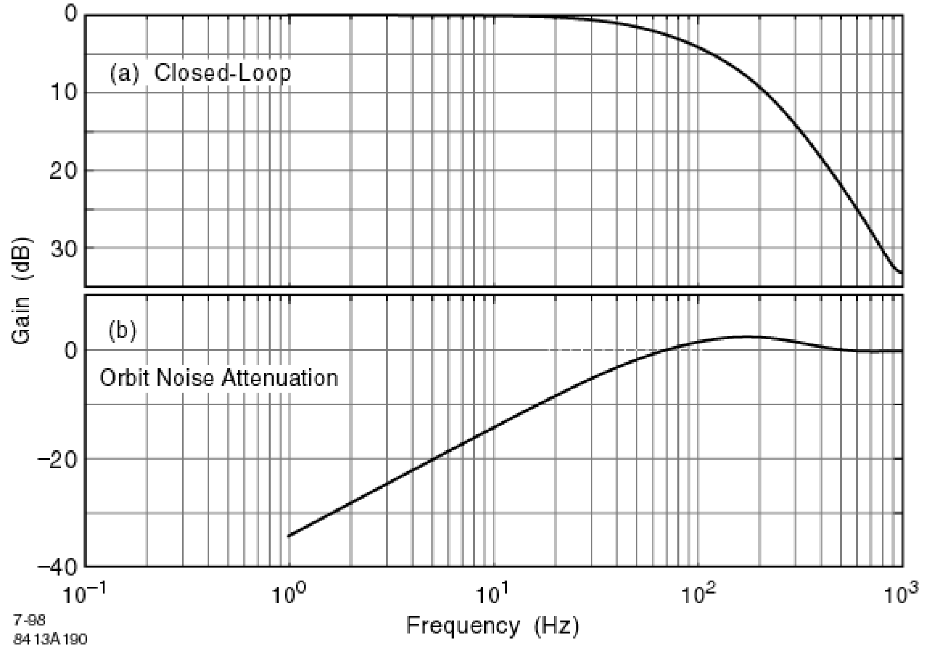
BPM Processing Remote Crate (1 of 4)



Orbit Feedback Performance



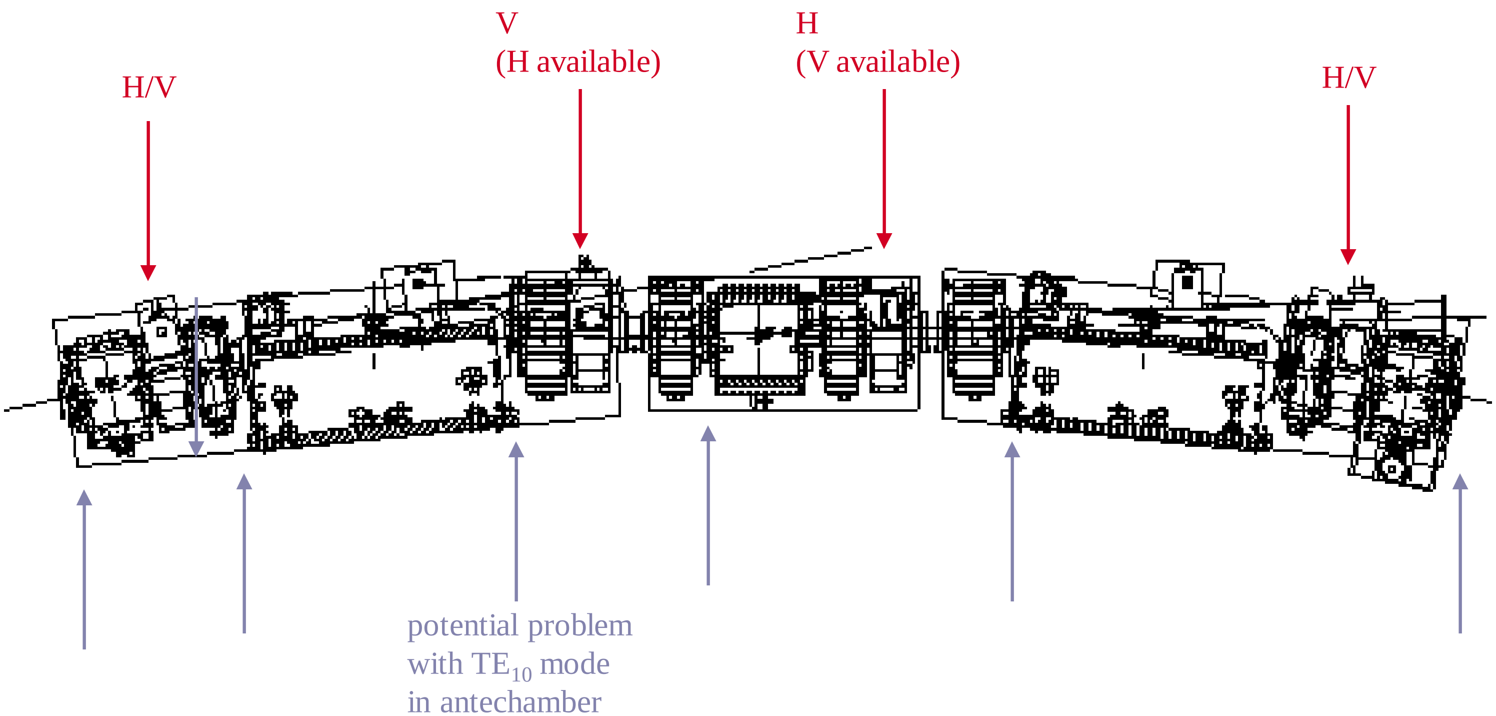




BPM and Orbit Corrector Locations

Corrector locations

(72 total; 54 H and 54 V correctors used for orbit feedbck)



BPM locations

(104 total; 90 used for orbit feedback)

BPM Processing and Orbit Feedback Performance Specifications

BPM Processing System		Orbit Feedback System	
Number BPMs	90	Number electron BPMs	90
Nominal beam current range	1-500 mA	Number photon BPMs	11 + future
First turn resolution (.025 nC bunch; .03 mA)	1.8 mm	Number correctors	54 H, 54 V
Turn-turn resolution (> 5mA)	12.7 μm	Closed-loop bandwidth (-3 dB)	100 Hz
Resolution for fdbk (2 kHz orb update; >5 mA)	1.1 μm (144-turn avg)	Component bandwidth	
Resolution for 1 s orbit averaging (>5 mA)	0.045 μm (>5 mA)	orbit monitor @ 2 kHz update	200 Hz
Resolution parameter (<5 mA; no multiplexing)	0.056 μm-mA/ √Hz	magnets	>1 kHz
Current dependence (for x2 ΔI); stability over 24 h	<3 μm	power supplies	500 Hz
Absolute BPM accuracy wrt quad center*	<100 μm	vacuum chamber	60 Hz H, 100 Hz V
Dynamic current range (for <10 μm turn-turn res.)	5 mA-500 mA (40 dB)	Stability goal (rms at BPMs)	<25 μm H,<5 μm V
Position range	± 1 cm V, ± 2 cm H		
RF button multiplexing	4:1		
Button mux switch period (min)	~10 μs / button		
Orbit update rate (max)	25 kHz (10 μs/button)		

BPM Processing and Orbit Feedback Component Development

RF-IF Converters

- 64 BPMs initially; 92 later
- Modify existing design for new RF frequency
- Considering commercial manufacturer

Digital IF Processors

- 8 ea. 8-channel modules (+ spares)
- Commercial vendor; 1st units received

Timing/Crate Driver Module

- 4 ea. + spares, SLAC design nearly complete

Remote Crate BPM Data Acquisition CPUs

- Power PC + 2 PMC slots (Synergy)
- RACEway link (PMC) to IF Processors (**160 MB/s**)
- Reflective memory link (PMC) to central crate

  >12 Mb/s for each of 4 crates

Orbit Feedback DSP

- Dual Power PC + 2 PMC slots (Synergy)
- Reflective memory link to 4 remote crates
- Fast Ethernet link (PMC) to corrector supply controllers (**100 Mb/s**)

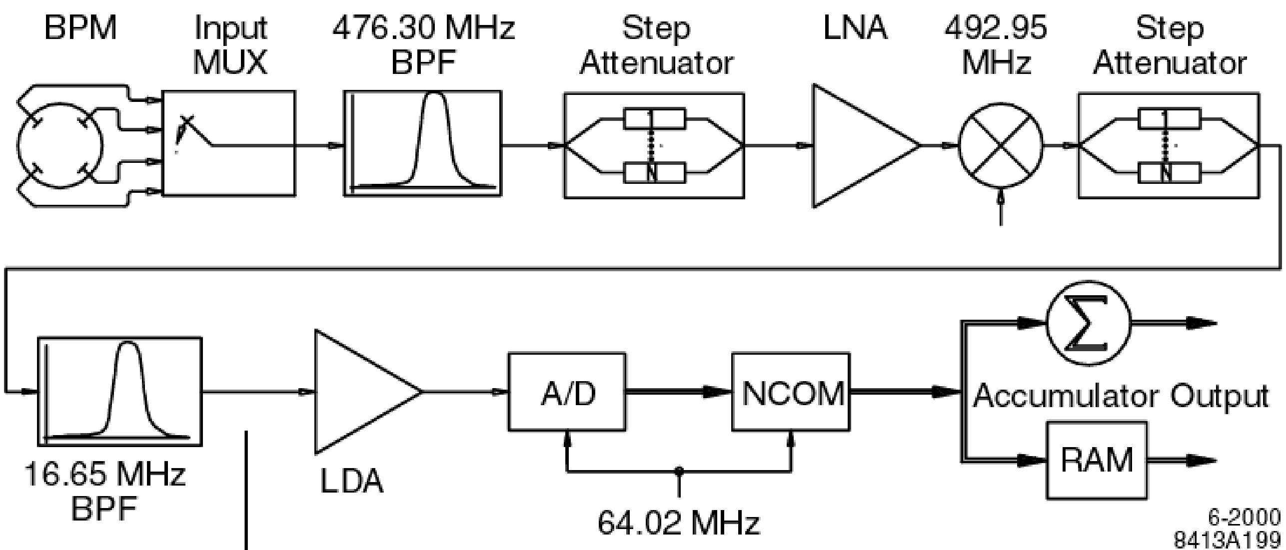
Fast Digital Power Supply Controllers

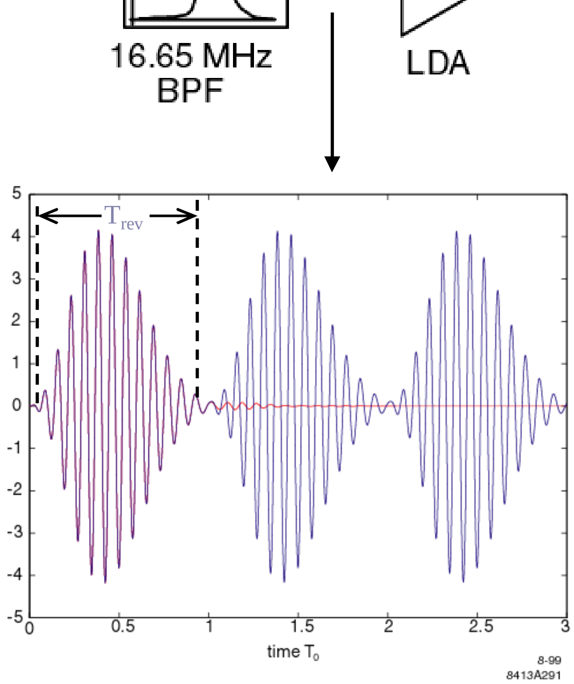
- 15 ea (+ spares) crate-based 8-channel controllers
- 4 kHz aggregate update rate with Fast Ethernet
- Digital regulation capability
- SLAC design

LO and Timing Generators

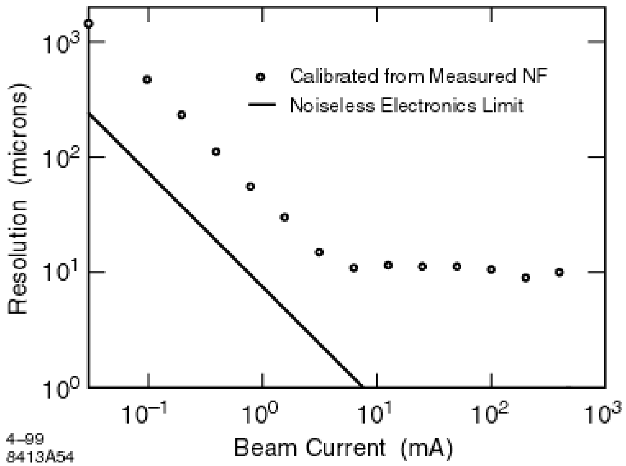
- Signals derived from 476.3 MHz MO
- Commercial low noise design

BPM Processor

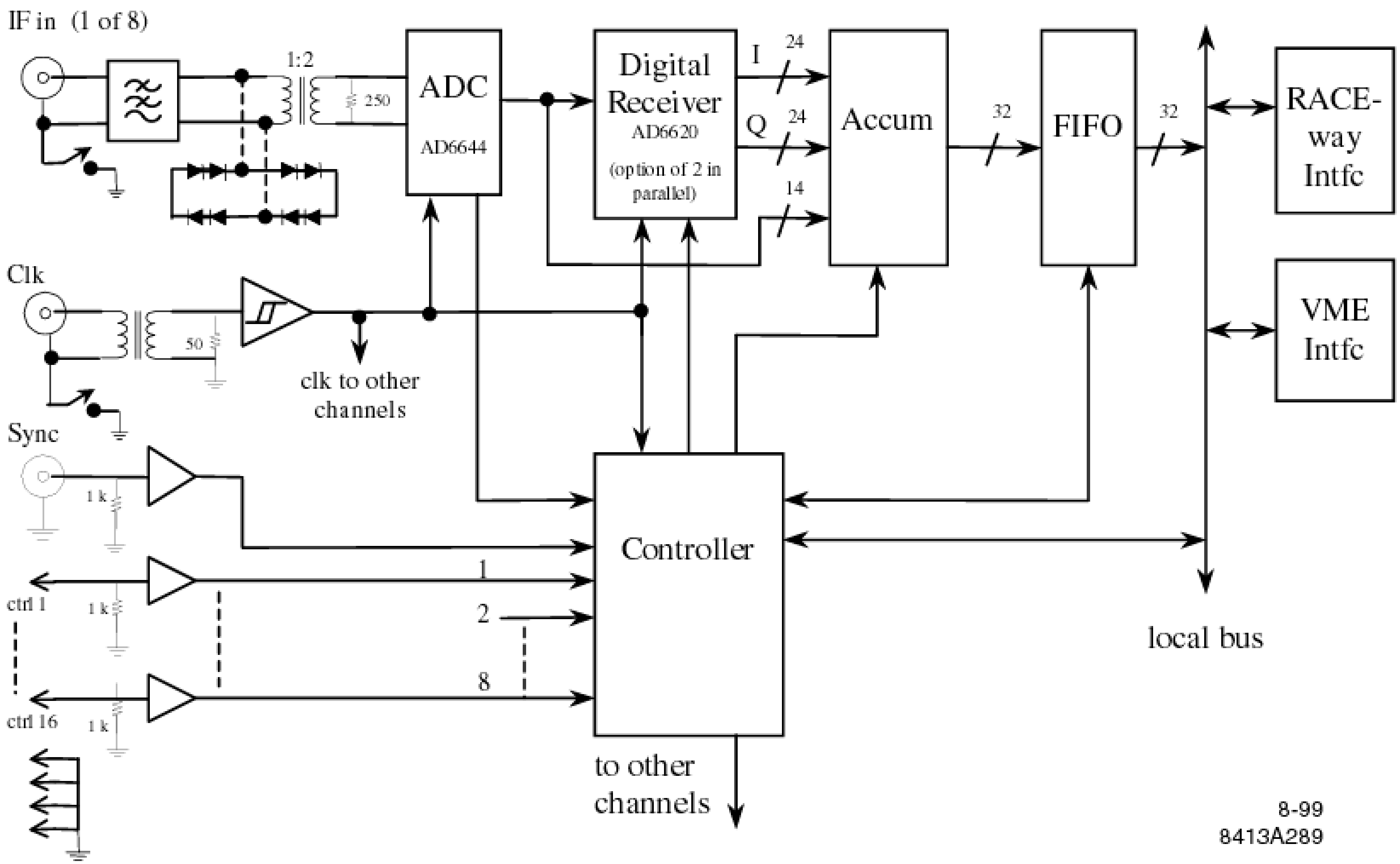




single-bunch response



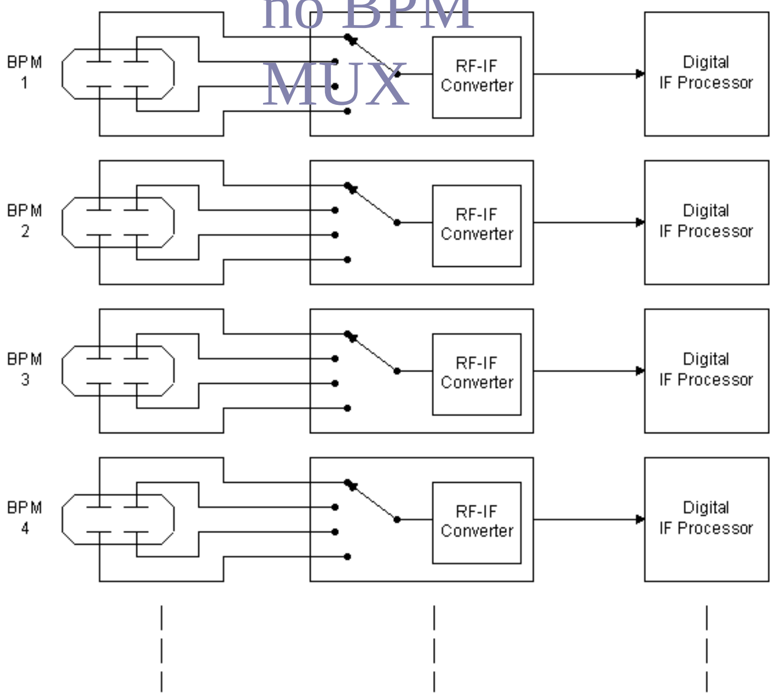
BPM Processing - IF Processor



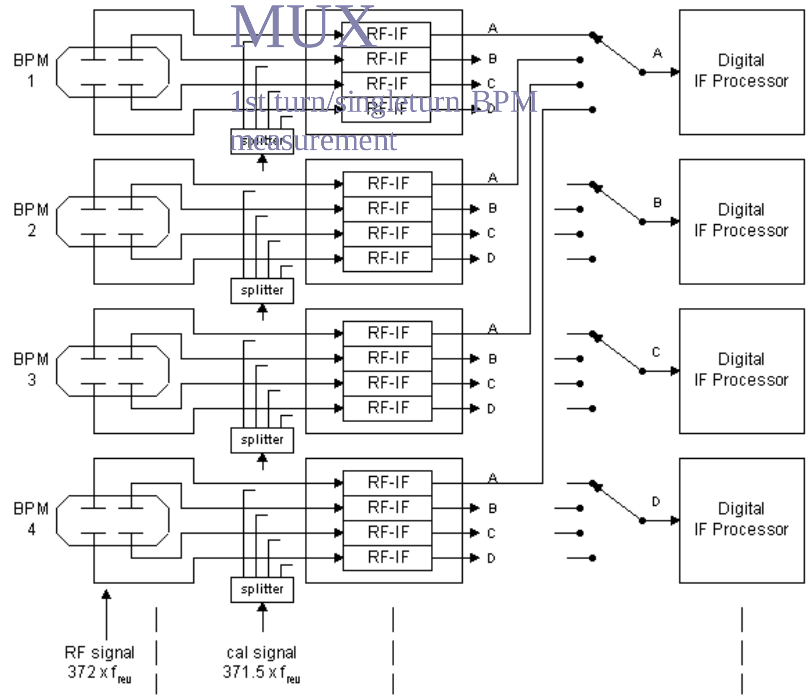
8-99
8413A289

BPM RF-IF Processor Options

4:1 button MUX
no BPM MUX

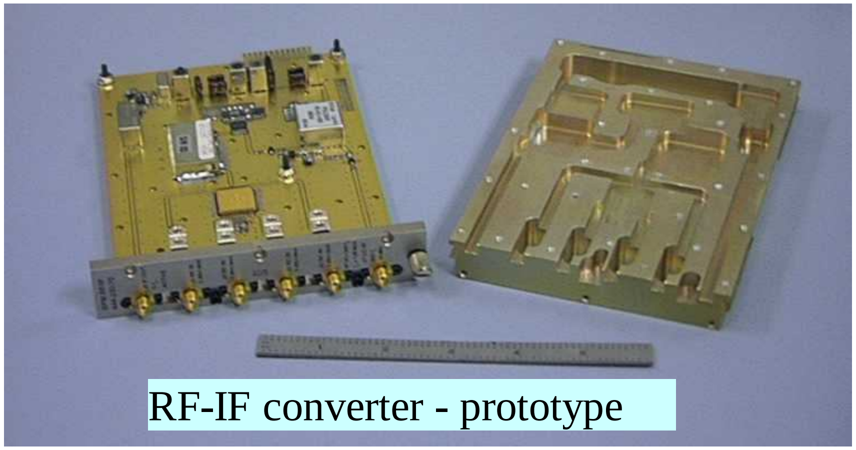


no button MUX
4:1 BPM MUX
1st turn/single turn BPM measurement

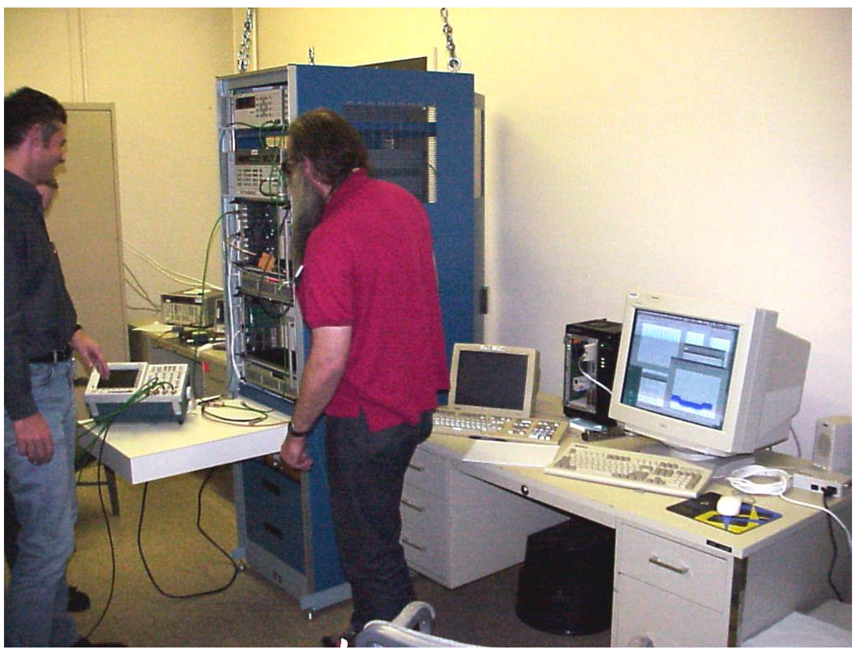


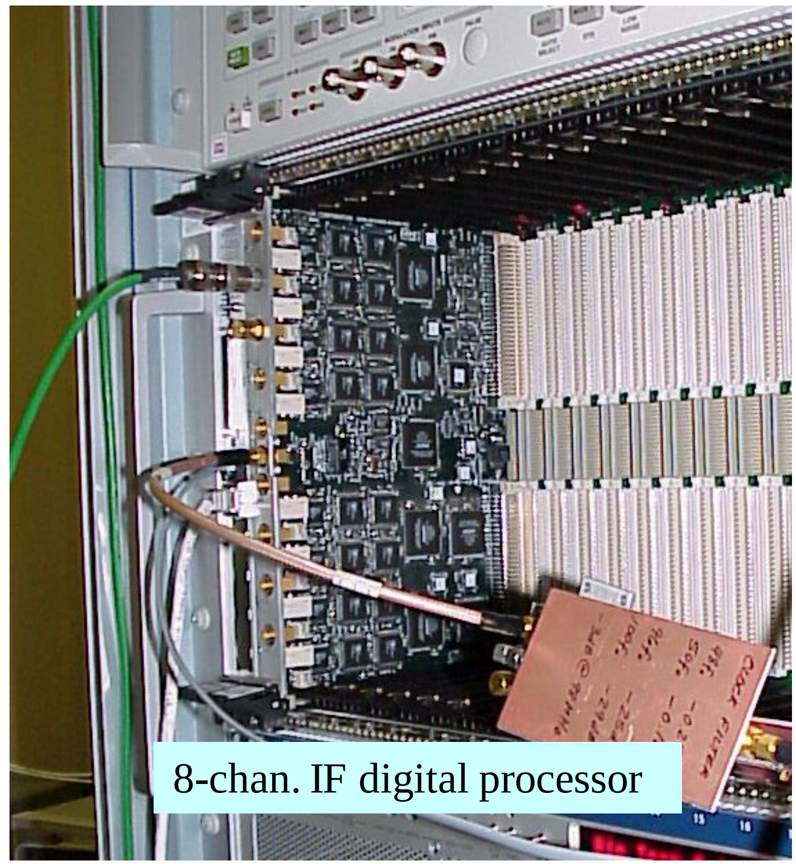
BPM Processing

# BPMs		90	
Resolution	1st turn:	1.8 mm	(0.03 mA)
	turn-turn:	13 µm	(> 5 mA)
	feedback:	1 µm	(160 avg)
Current range		5-500 mA	(<13 µm turn-turn res)
Current dependency		< 3 µm	
Orbit acquisition rate		2-4 kHz for feedback	(~25 kHz max)



RF-IF converter - prototype

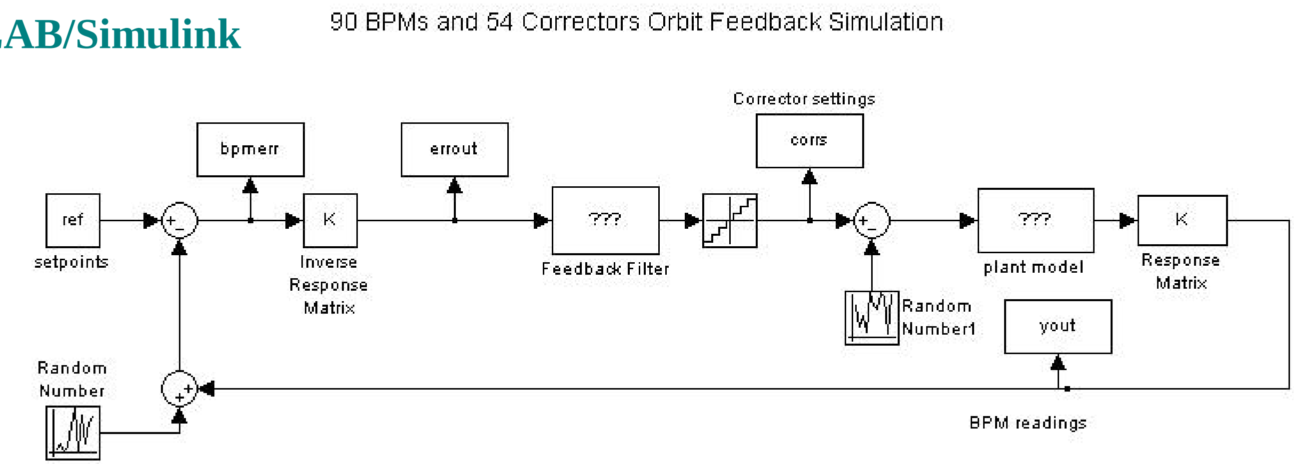


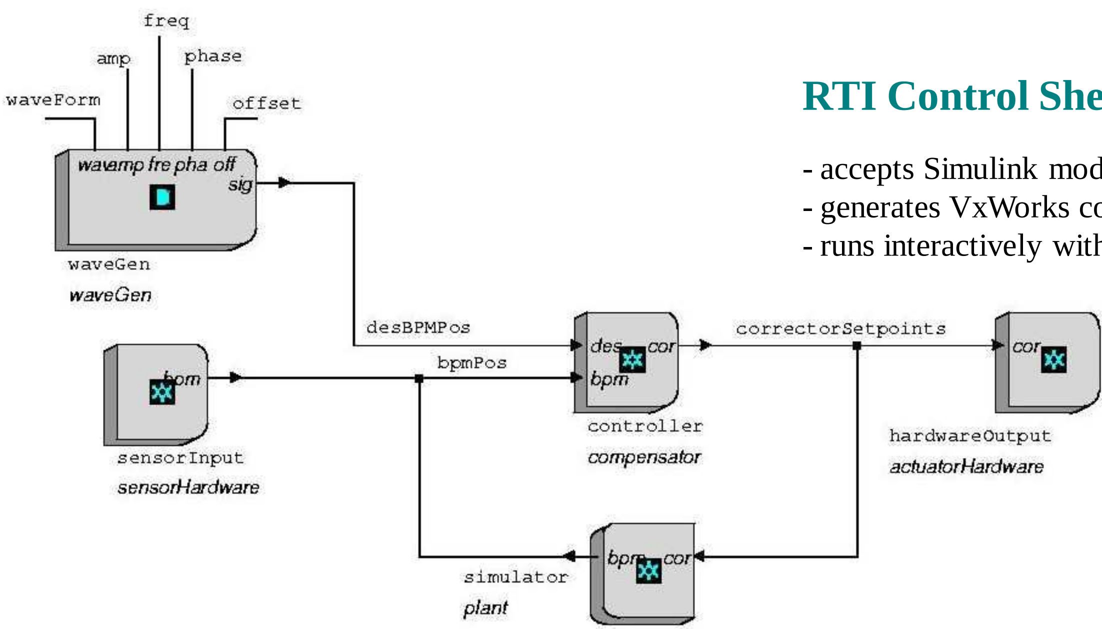


8-chan. IF digital processor

Orbit Feedback Modeling and Programming

MATLAB/Simulink





RTI Control Shell

- accepts Simulink models, other representations
- generates VxWorks code
- runs interactively with model and/or real world

Machine Protection Systems

Vacuum Interlock

- PLC 1
- ~160 vacuum chamber water flow switches
- ~24 ion gauges
- ~310 thermal switches
- 12 BL Vacuum OK summaries
- Enables RF, ring isolation valves + stoppers
- Expand existing system

Chamber Temperature Monitor

- PLC 3
- ~375 thermocouples (30-200^{o}C ± 1^{o}C)
- ~16 RTDs (10- 70^{o}C ± 1^{o}C)
- ~960 chan/s measurement rate
- Generates alarms, status for Control System
- New system, commercially available

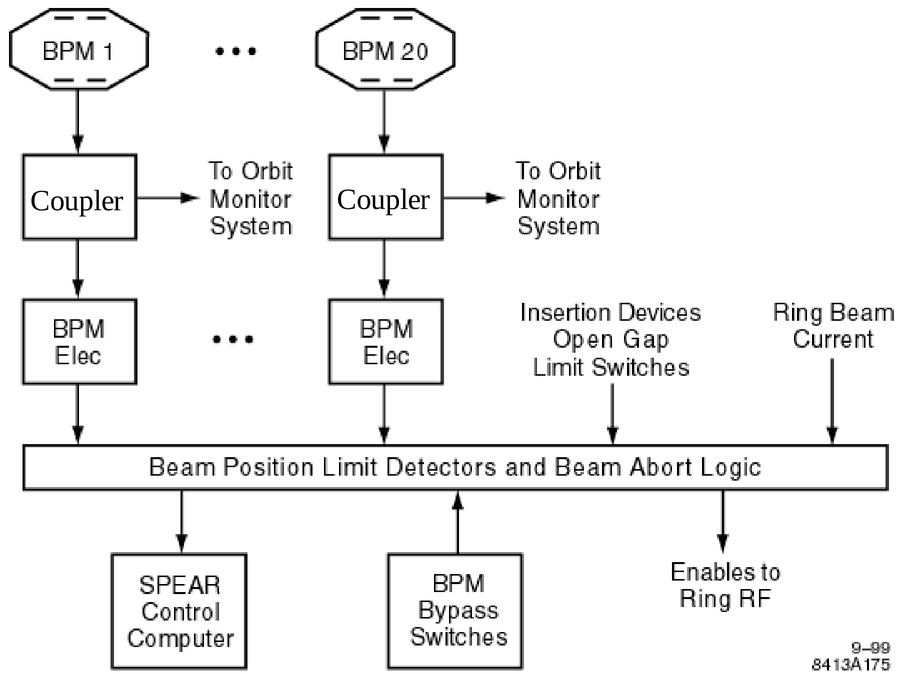
Magnet Cooling Interlock

- PLC 2
- ~20 water flow switches
- ~1550 thermal switches/~264 interlock circuits
- Enable magnet power supplies
- Expand existing system

Orbit Interlock

- Active for beam current >20 mA
  - less if beam lines open
- 20 BPMs (in beam line areas)
  - future expansion: 2 per new ID; 30 total
- New system design
  - BPM processors: commercial or SLAC design
  - BPLD and Beam Abort: new design, VME components

Orbit Interlock



Specifications

Number of BPMs	20
Processing frequency	476.3 MHz
Beam current range (nom)	5-500 mA
Resolution (>5 mA)	<50 µm
Accuracy (wrt quad center)	<100 µm
Dynamic range (intensity)	>60 dB
Channel isolation	>60 dB
Beam abort time (via RF system)	<1 ms

Interlock Trip Criterion (>20 mA)

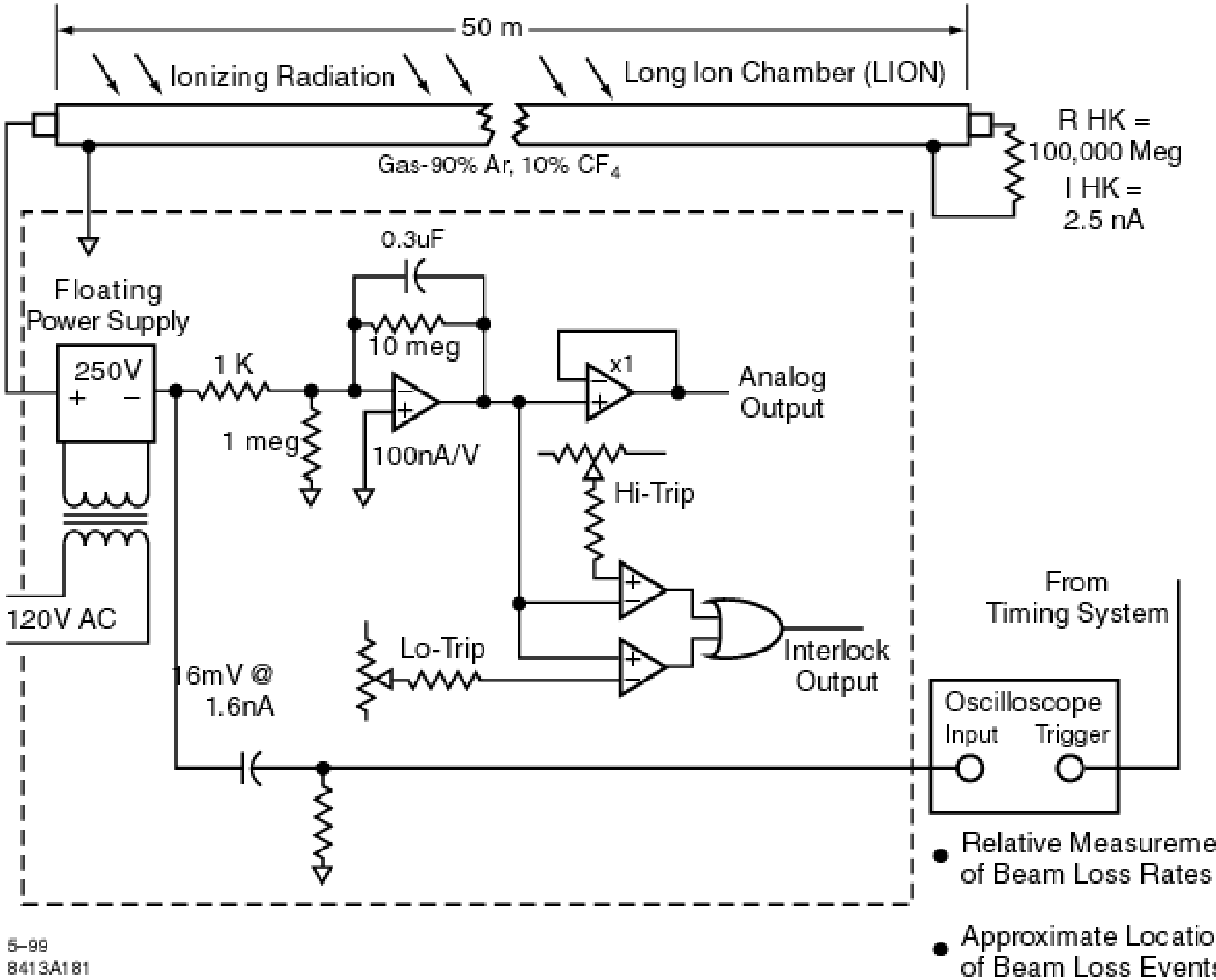
ID vertical: $\frac{|y|}{2.45\,\mathrm{mm}} + \frac{|y'|}{0.49\,\mathrm{mrad}} < 1$ **ID horizontal:** $\frac{|x|}{5.0\,\mathrm{mm}} + \frac{|x'|}{1.1\,\mathrm{mrad}} < 1$

(uses 2 ea ID straight BPMs per ID, 3 m apart ⇒ 0.73 mm |y| for angle trip)

Dipole BL vert: $|y| < 2.45$ mm **Dipole BL hor:** $|x| < 5$ mm

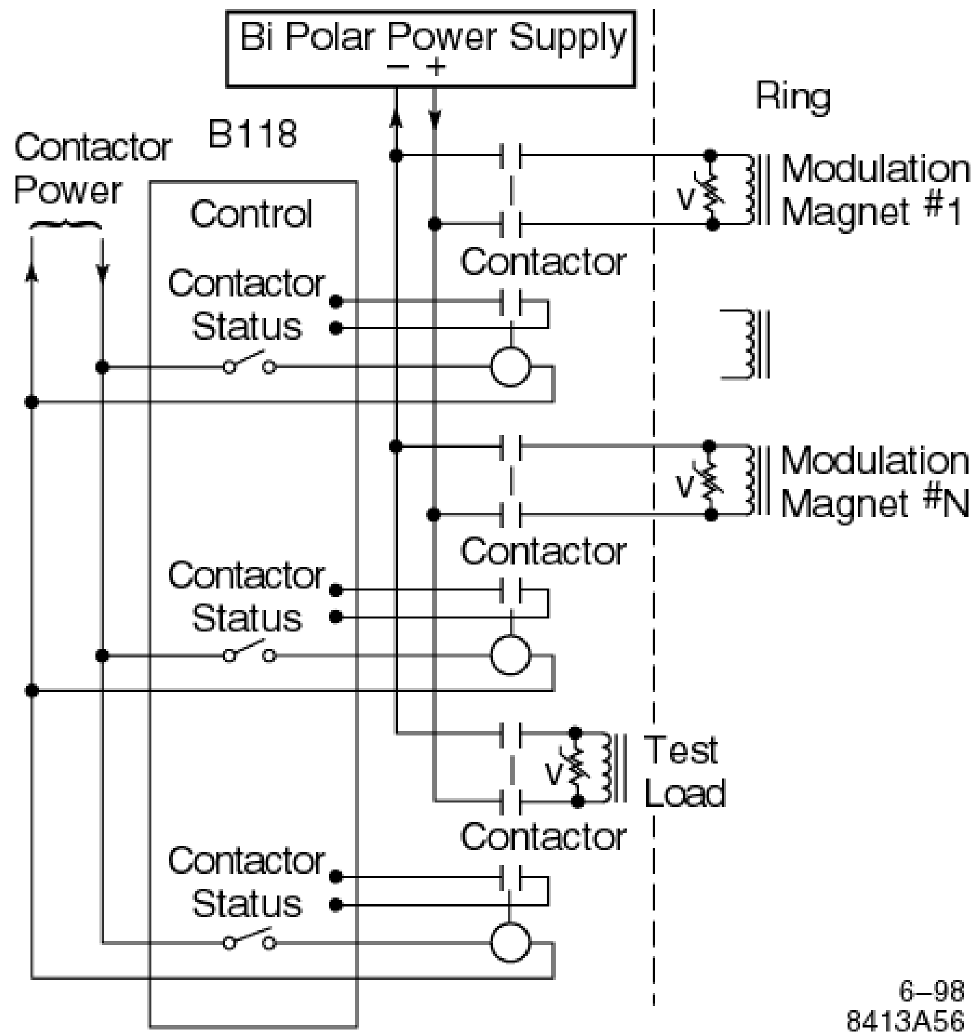
(uses upstream ID BPM and downstream dipole BPM per dipole source point)

Beam Containment System
Long Ion Chamber (LION)



- Relative Measurement of Beam Loss Rates
- Approximate Location of Beam Loss Events

Quadrupole Modulation System



DCCT

Parametric Current Transformer (Bergoz)

1 A full scale

0.5 μA resolution (1s integration)

Dynamic range > $2x10^7$

Absolute accuracy < 0.05%

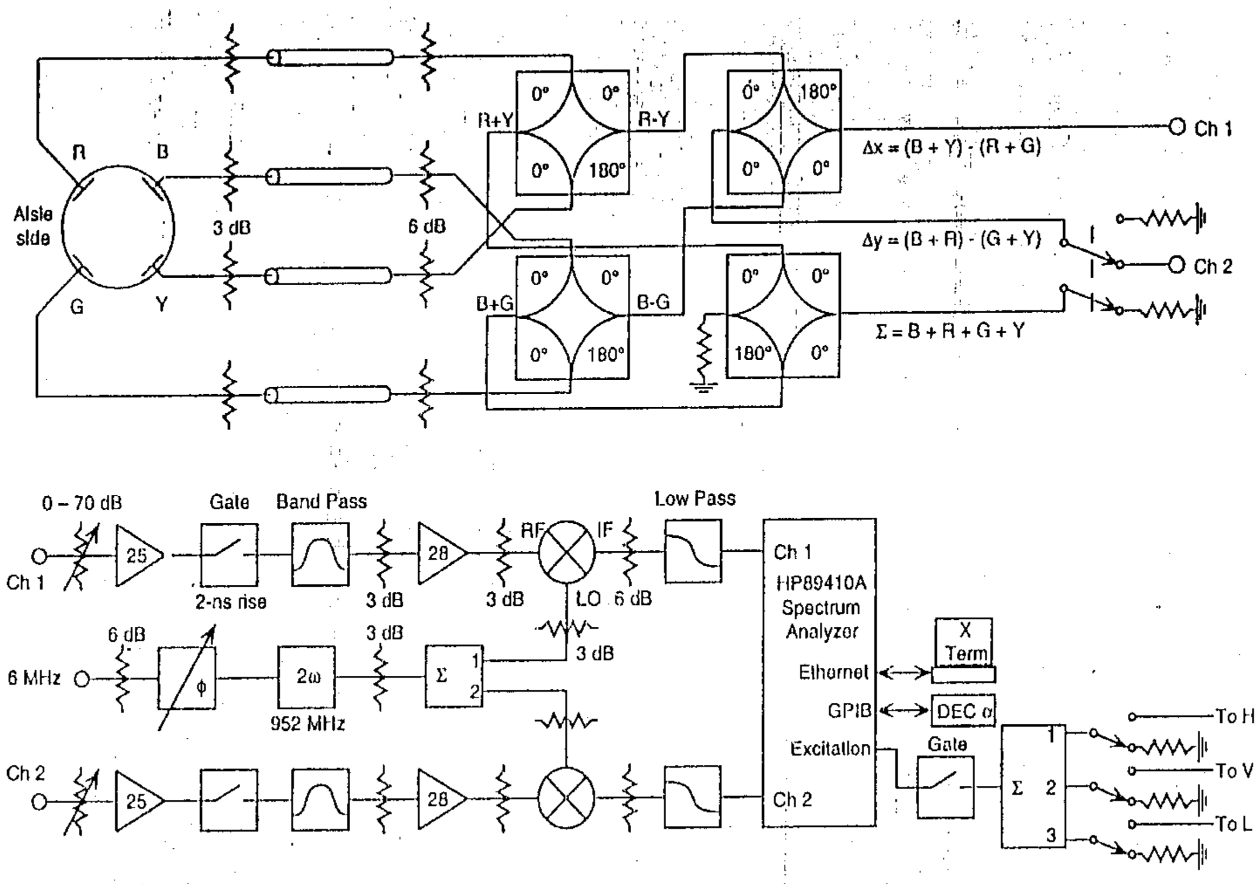
Linearity error < 0.01%

DC -100 kHz

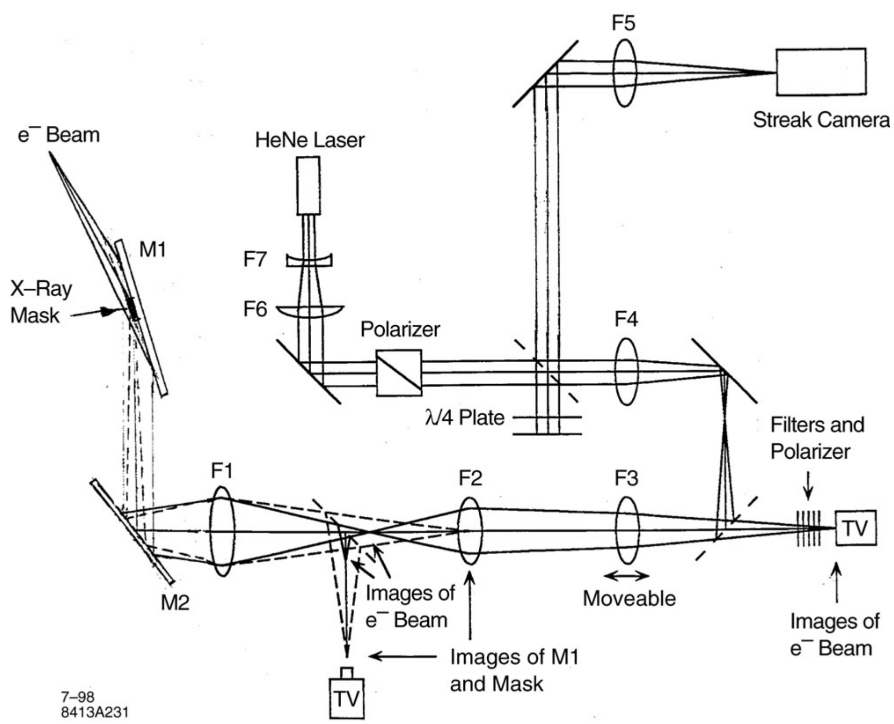
Output +/- 10V bipolar

113 or 175 mm ID

Tune Monitor



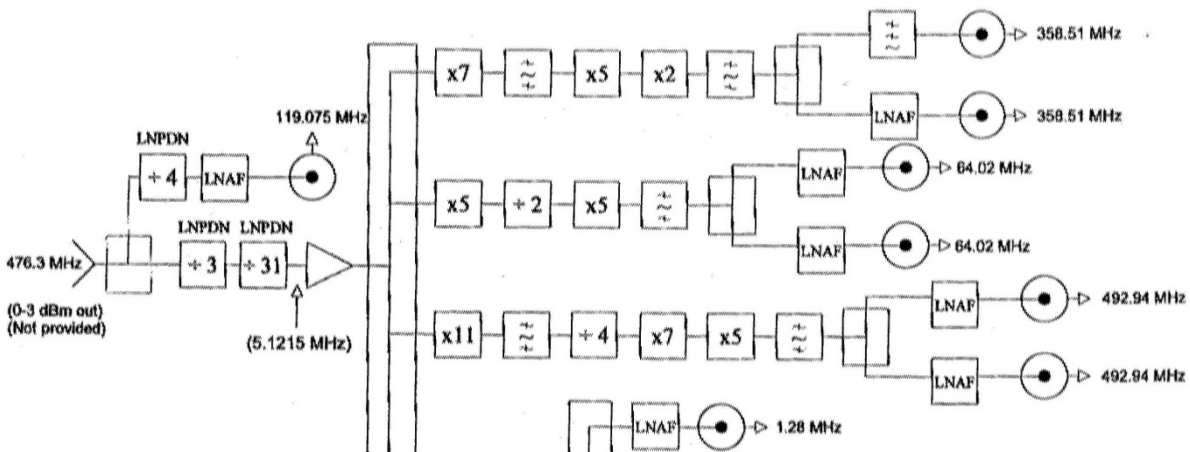
Synchrotron Light Monitor



Parameter	Value
Radius of curvature in dipole r	7.86 m
Critical energy in dipole *Ec*	7.62 keV
Critical wavelength in dipole lc	0.163 nm
Measurement wavelength l	210 nm
Opening angle (1/g) at lc	0.17 mrad
Opening angle at l for both polarizations	1.87 mrad
Opening angle at l for horizontal polarization	1.12 mrad
Diffraction spot size sd	15 µm
Electron beam size s*x*	183 µm
Electron beam size s*y*	51 µm
s*y* /sd	3.41
Vertical image size simage (1:1 image)	53 µm
simage /s*y*	1.04

SPEAR 3 Timing and RF Signal Generator System

SPEAR RF:	$f_{SPrf} = 372 \times f_{SPrev} = 476.300$ MHz	**BPM LO:**	$f_{LO} = 385 \times f_{SPrev} = 492.935$ MHz
Booster RF:	$f_{Brf} = 280 \times f_{SPrev} = 358.505$ MHz	**BPM IF:**	$f_{IF} = 13 \times f_{SPrev} = 16.645$ MHz
SPEAR revolution freq:	$f_{SPrev} = 1.2804$ MHz	**IF digitizing clock:**	$f_{IFclk} = 50 \times f_{SPrev} = 64.020$ MHz
		Streak camera clock:	$f_{SC} = f_{SPrf}/4 = 93 \times f_{SPrev} = 119.075$ MHz

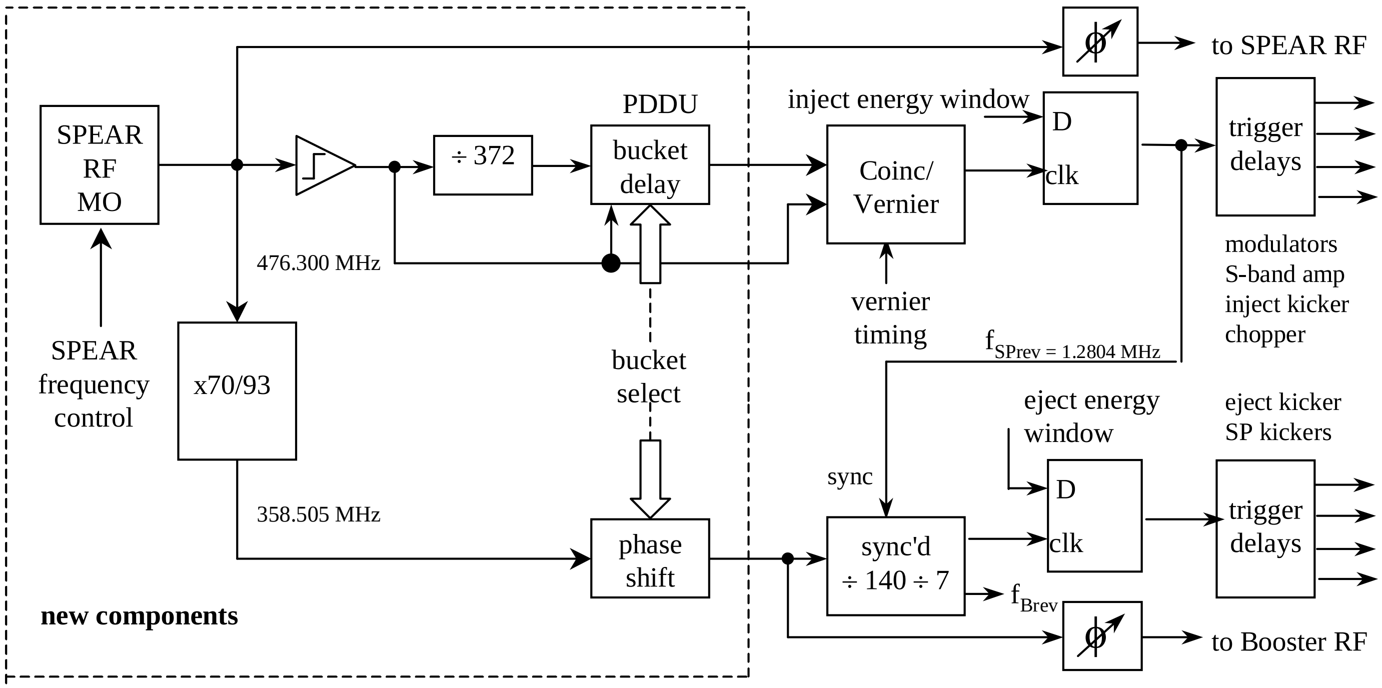


SPECIFICATIONS:
- All Outputs +10 dBm
- Harmonics -30 dBc
- Subharmonics -60 dBc
- Phase Noise Floor Intrinsic at Output

FREQ.	PHASE NOISE AT 20 kHz
1.28 MHz	-168 dBc/Hz
16.64	-145 dBc/Hz
64.02	-145 dBc/Hz
119.075	-168 dBc/Hz
358.51	-130 dBc/Hz
492.94	-128 dBc/Hz

Wenzel, Inc.

Injection Timing System



- Single-bunch filling
- Phase-lock Booster RF to SPEAR RF:

$$C_{Boo}/C_{SPEAR} = 4/7$$

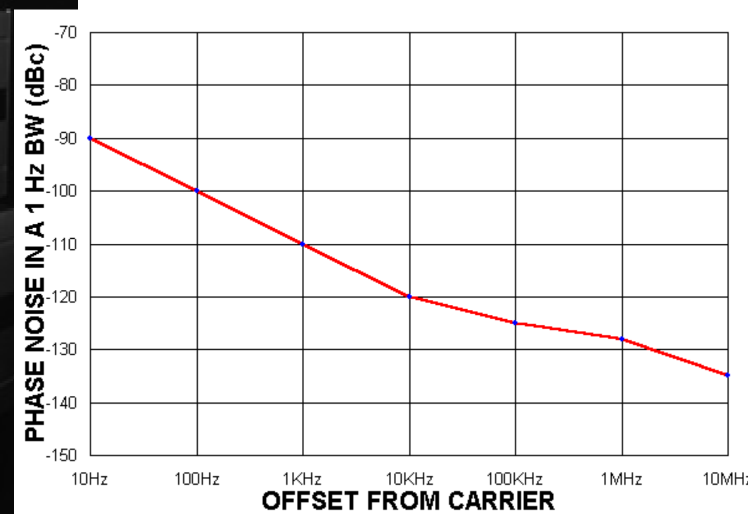
$$h_{SP} = 372 \quad h_{Boo} = 160 \quad \Rightarrow f_{Boo}/f_{SPEAR} = 70/93$$

$$f_{SP} = 476.300 \text{ MHz} \;\Rightarrow\; f_{Boo} = 358.505 \text{ MHz}$$

- Discrete bucket-dependent phase shift of f_{Boo}:

$$\phi_{N+1} = \phi_N + 360^o \times 70/93 = \phi_N + 271^o = \phi_N - 89^o$$

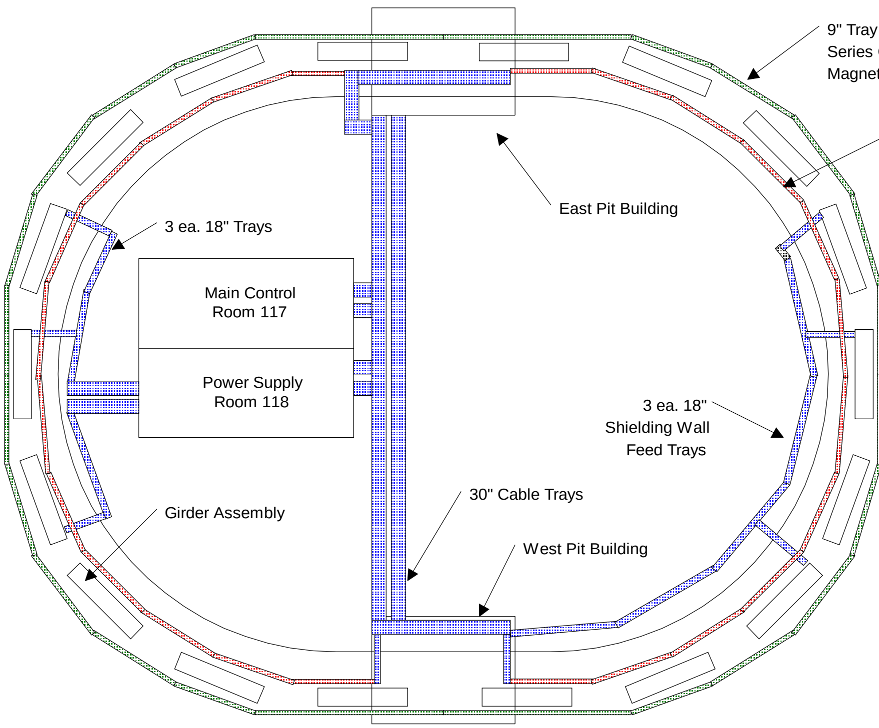
SPEAR 3 Master Oscillator

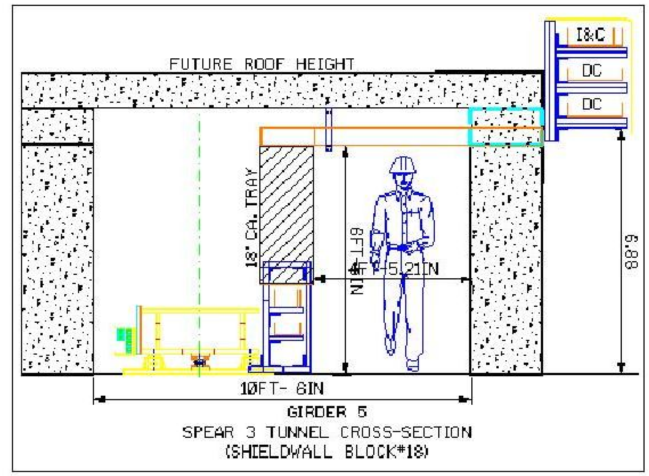


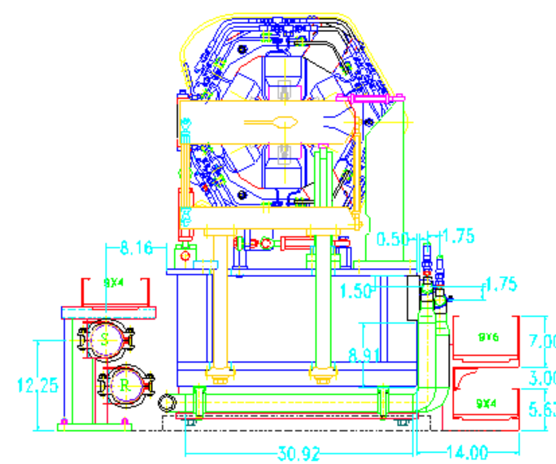
- 1-500 MHz, DDS-based
- 0.2 Hz step resolution
- Phase-continuous frequency switching
- Stability: $3\text{x}10^{-9}$/day, 10^{-6}/yr, $\pm 10^{-8}$/0-50^{o}C
- 0.057^{o} integrated phase noise, 0.5 Hz-15 kHz
- GPIB control

Cable Tray Routes

SPEAR3 CABLE TRAY PLAN





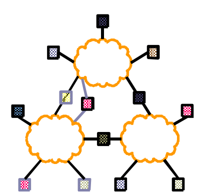


30" Tray from Buildings 117/118 to Ring
18" Tray to Shielding Wall Feed Points
12" Tray in Matching Cells
9" Tray adjacent to girders
6" Tray adjacent to Insertion Devices

*outer shielding not shown

UNIT – IV

Distributed Control Systems



Distributed Control Systems

- Collection of hardware and instrumentation necessary for implementing control systems
- Provide the infrastructure (platform) for implementing advanced control algorithms

History of Control Hardware

- Pneumatic Implementation:
  - *Transmission*: the signals transmitted pneumatically are slow responding and susceptible to interference.
  - *Calculation*: Mechanical computation devices must be relatively simple and tend to wear out quickly.

History (cont.)

- Electron analog implementation:
  - *Transmission*: analog signals are susceptible to noise, and signal quality degrades over long transmission line.
  - *Calculation*: the type of computations possible with electronic analog devices is still limited.

History (cont.)

- Digital Implementation:
  - *Transmission*: Digital signals are far less sensitive to noise.
  - *Calculation*: The computational devices are digital computers.

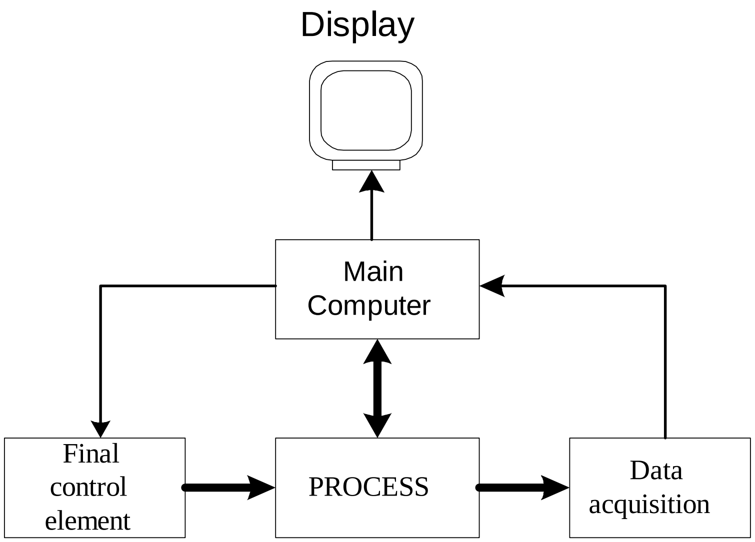
Advantages of Digital System

- Digital computers are more flexible because they are programmable and no limitation to the complexity of the computations it can carry out.
- Digital systems are more precise.
- Digital system cost less to install and maintain
- Digital data in electronic files can be printed out, displayed on color terminals, stored in highly compressed form.

Computer Control Networks

1. PC Control:

- Good for small processes such as laboratory prototype or pilot plants, where the number of control loops is relatively small



Computer Control Networks

2. Programmable Logic Controllers:

- specialized for non-continuous systems such as batch processes.

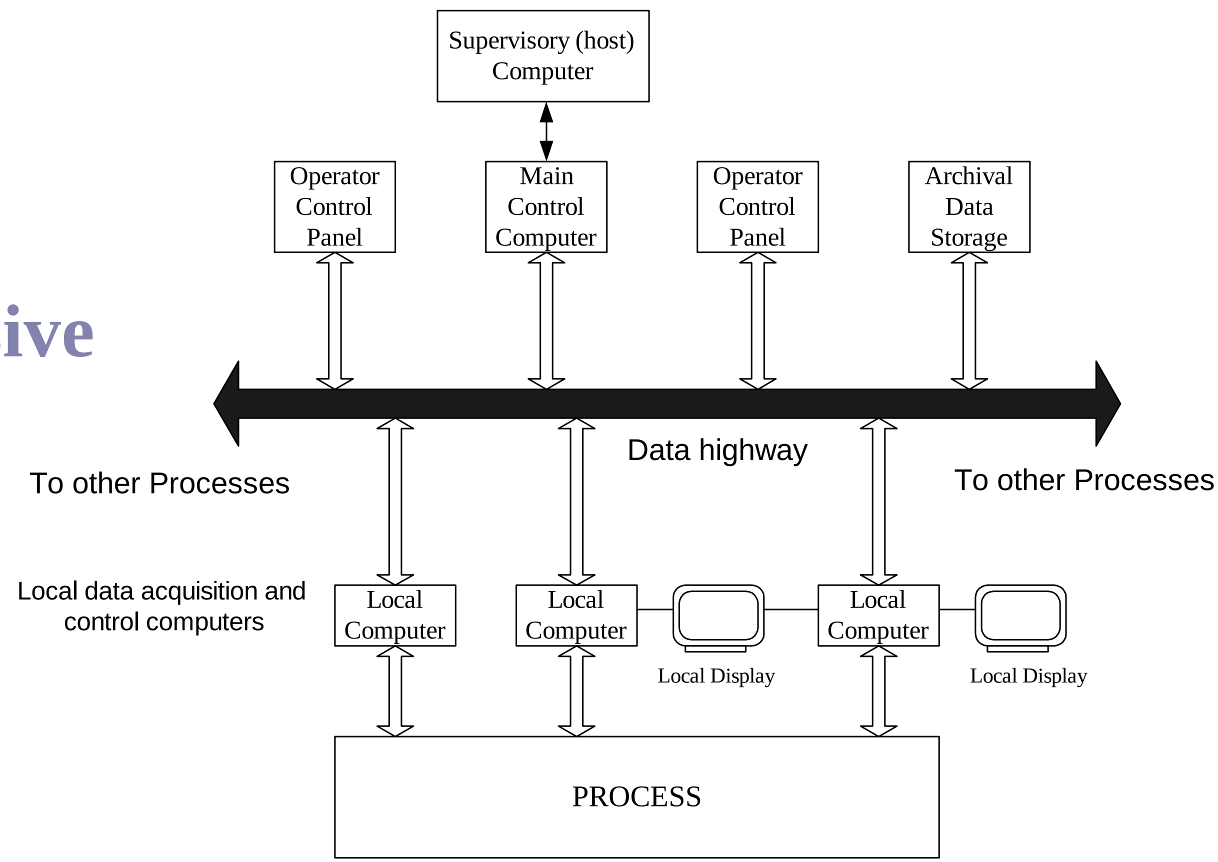
- It can be used when interlocks are required; e.g., a flow control loop cannot be actuated unless a pump has been turned on.

- During startup or shutdown of continuous processes.

Computer Control Networks

3. DCS

- **Most comprehensive**



DCS Elements-1

- **Local Control Unit**: This unit can handle 8 to 16 individual PID loops.
- **Data Acquisition Unit**: Digital (discrete) and analog I/O can be handle.
- **Batch Sequencing Unit**: This unit controls a timing counters, arbitrary function generators, and internal logic.
- **Local Display**: This device provides analog display stations, and video display for readout.
- **Bulk Memory Unit:** This unit is used to store and recall process data.

DCS Elements-2

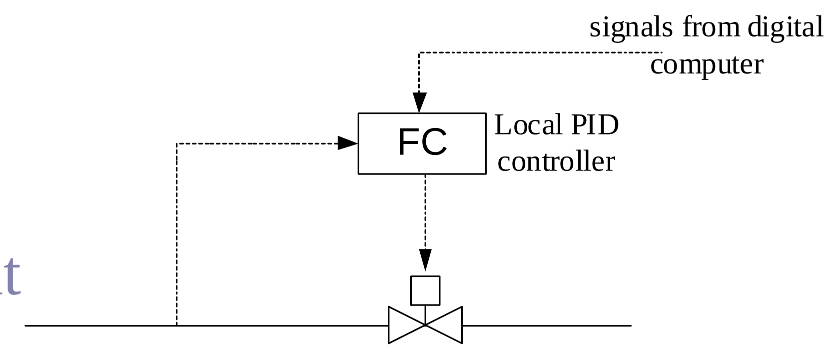
- **General Purpose Computer** : This unit is programmed by a customer or third party to perform optimization, advance control, expert system, etc
- **Central Operator Display**: This unit typically contain several consoles for operator communication with the system, and multiple video color graphics display units
- **Data Highway** : A serial digital data transmission link connecting all other components in the system. It allow for redundant data highway to reduce the risk of data loss
- **Local area Network** *(LAN)*

Advantages of DCS

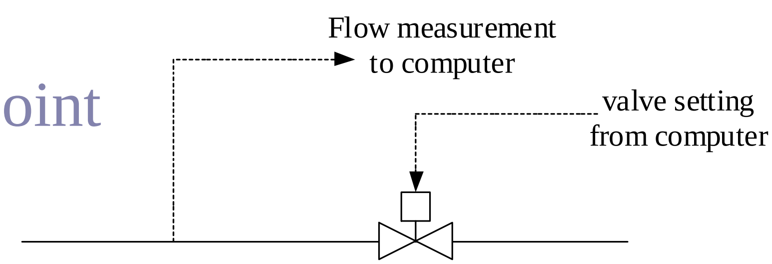
- Access a large amount of current information from the data highway.
- Monitoring trends of past process conditions.
- Readily install new on-line measurements together with local computers.
- Alternate quickly among standard control strategies and readjust controller parameters in software.
- A sight full engineer can use the flexibility of the framework to implement his latest controller design ideas on the host computer.

Modes of Computer control

1. Manual
2. Automatic
   - PID with local set point
3. Supervisory
   - PID with remote set point (supervisory)
4. Advanced



Supervisory Control mode



Direct digital Control mode

Additional Advantage

- Digital DCS systems are more flexible. Control algorithms can be changed and control configuration can be modified without having rewiring the system.

Categories of process information

Type	*Example*
1. Digital	Relay, Switch Solenoid valve Motor drive
2. Generalized digital	Alphanumerical displays
3. Pulse	Turbine flow meter Stepping motor
4. Analog	Thermocouple or strain gauge (mill volt) Process instrumentation (4-20 am) Other sensors (0-5 Volt)

Interface between digital computer and analog instruments

- (A/D) Transducers convert analog signals to digital signals. (Sensor $\longrightarrow$ Computer)

- (D/A) Transducers convert digital signals to analog signals. (Computer $\longrightarrow$ Valve)

Data resolution due to digitization

- Accuracy depends on resolution.
- Resolution depends on number of bits:

Resolution = signal range $\times$ $1/(2^m - 1)$

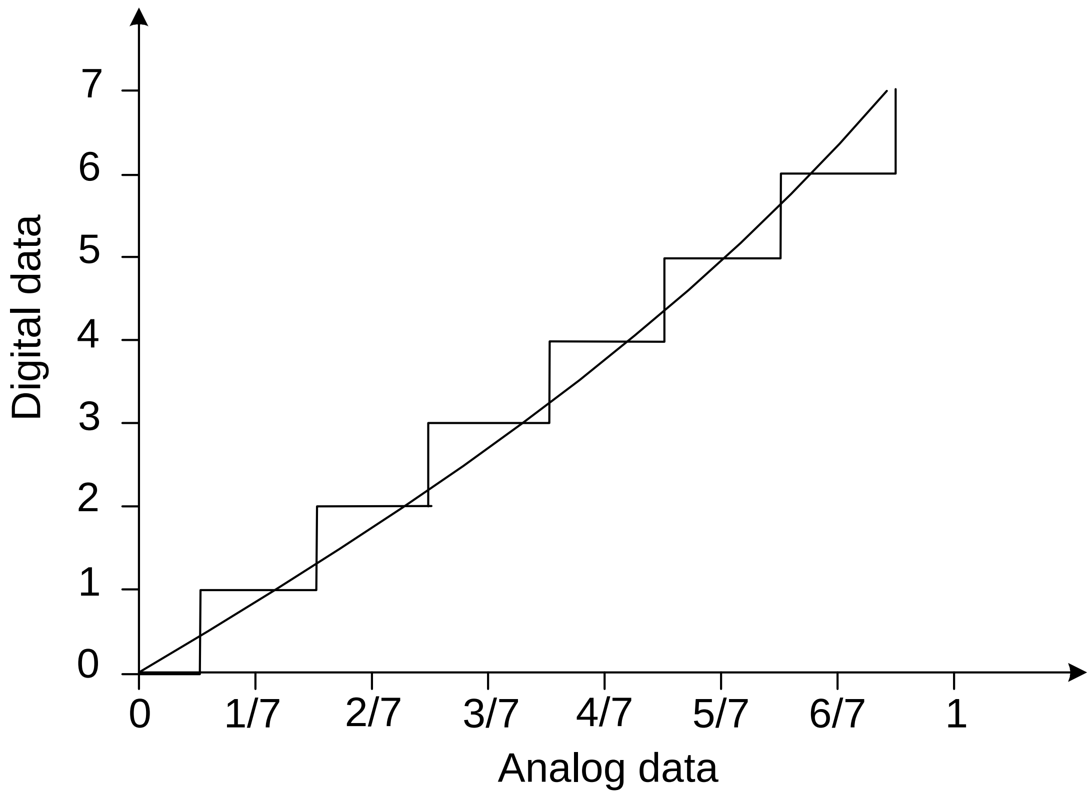
m = number of bits used by the digitizer (A/D) to represent the analog data

Data Resolution

- Signal = 0 - 1 Volt, 3 bit digitizer:

Binary representation	Digital Equivalent	Analog equivalent	Analog range covered
0 0 0	0	0	0 to 1/14
0 0 1	1	1/7	1/14 to 3/14
0 1 0	2	2/7	3/14 to 5/14
0 1 1	3	3/7	5/14 to 7/14
1 0 0	4	4/7	7/14 to 9/14
1 0 1	5	5/7	9/14 to 11/14
1 1 0	6	6/7	11/14 to 13/14
1 1 1	7	1	13/14 to 14/14

Data Resolution



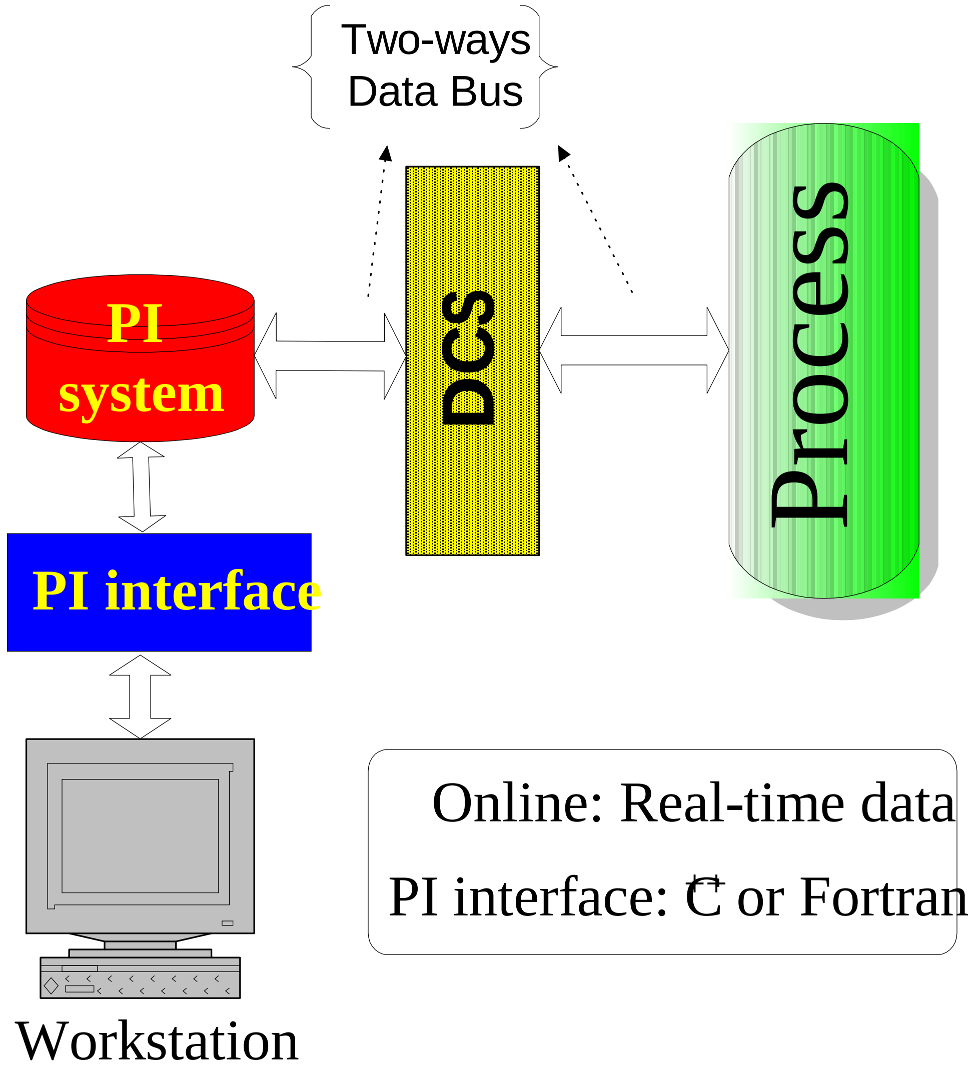
Utilization of DCS

- DCS vendor job:
  - installation
- Control Engineer Job:
  - Configuration
- Built-in PID control:
  - How to Tune the PID control?

Utilization of DCS

- Implementation of advanced control:
  - Developed software for control algorithms, DMC, Aspen, etc.
  - Control-oriented programming language supplied by the DCS vendors.
  - Self-developed programs using high-level programming languages (Fortran, C++)

Advanced control topology



DCS Vendors

- Honeywell
- Fisher-Rosemont
- Baily
- Foxboro
- Yokogawa
- Siemen

Unit V

INTERFACES IN DCS

Sections:

1. Process Industries vs. Discrete Manufacturing Industries
2. Continuous vs. Discrete Control
3. Computer Process Control

Industrial Control - Defined

The automatic regulation of unit operations and their associated equipment as well as the integration and coordination of the unit operations into the larger production system

- Unit operation
  - Usually refers to a manufacturing operation
  - Can also apply to material handling or other equipment

Process Industries vs. Discrete Manufacturing Industries

- Process industries
  - Production operations are performed on amounts of materials
  - Materials: liquids, gases, powders, etc.
- Discrete manufacturing industries
  - Production operations are performed on quantities of materials
  - Parts, product units

Definitions: Variable and Parameters

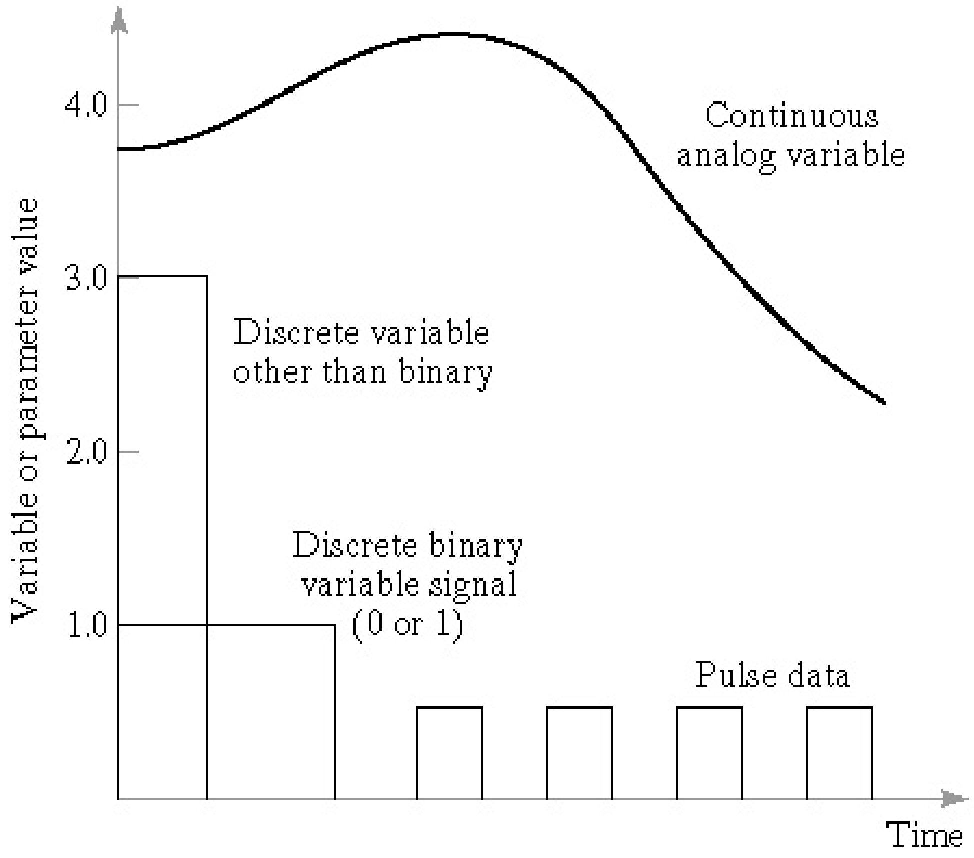
- Variables - outputs of the process
- Parameters - inputs to the process
- Continuous variables and parameters - they are uninterrupted as time proceeds
  - Also considered to be analog - can take on any value within a certain range
    - They are not restricted to a discrete set of values
- Discrete variables and parameters - can take on only certain values within a given range

Discrete Variables and Parameters

Categories:

- Binary - they can take on either of two possible values, ON or OFF, 1 or 0, etc.
- Discrete other than binary - they can take on more than two possible values but less than an infinite number of possible values
- Pulse data - a train of pulses that can be counted

Continuous and Discrete Variables and Parameters



Types of Control

- Just as there are two basic types of variables and parameters in processes, there are also two corresponding types of control:
  - Continuous control - variables and parameters are continuous and analog
  - Discrete control - variables and parameters are discrete, mostly binary discrete

Continuous Control

- Usual objective is to maintain the value of an output variable at a desired level
  - Parameters and variables are usually continuous
  - Similar to operation of a feedback control system
  - Most continuous industrial processes have multiple feedback loops
- Examples of continuous processes:
  - Control of the output of a chemical reaction that depends on temperature, pressure, etc.
  - Control of the position of a cutting tool relative to workpart in a CNC machine tool

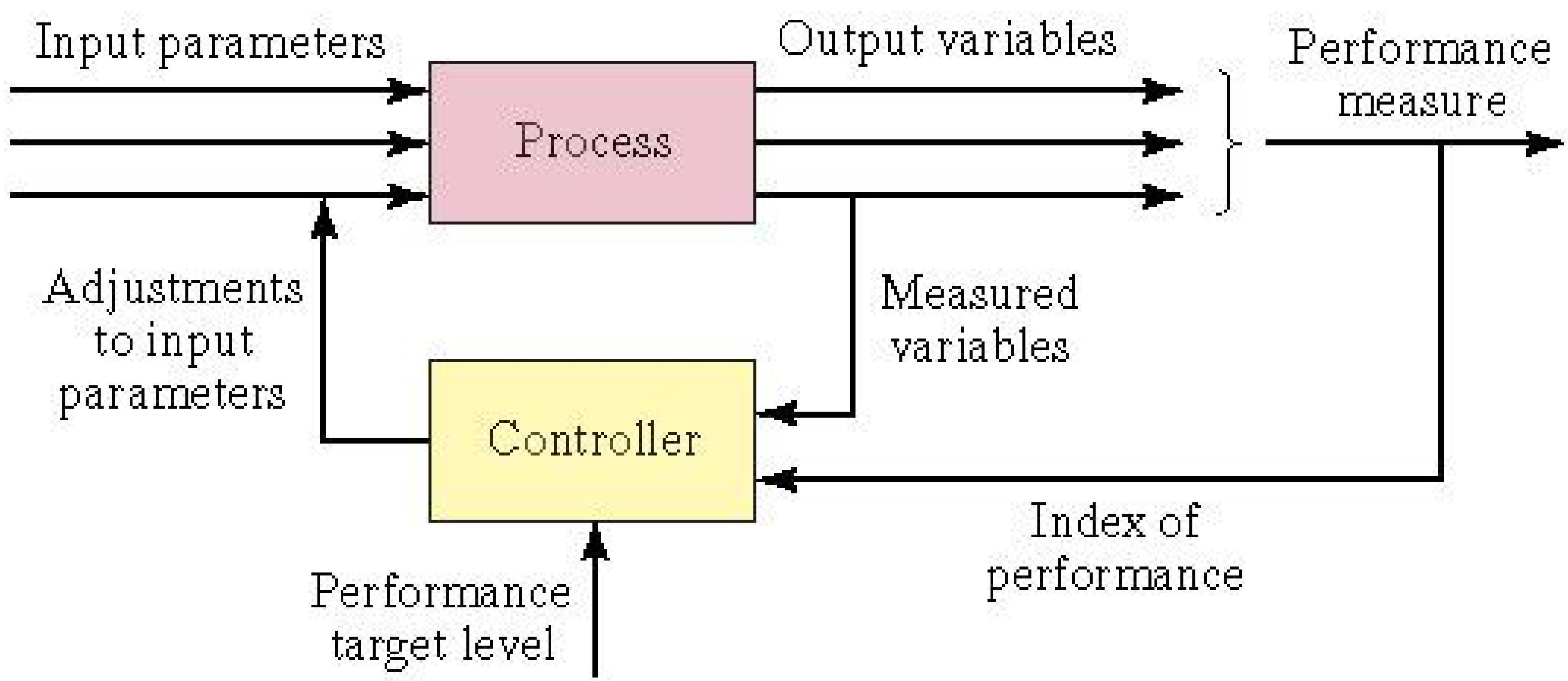
Types of Continuous Process Control

- Regulatory control
- Feedforward control
- Steady-State optimization
- Adaptive control
- On-line search strategies
- Other specialized techniques
  - Expert systems
  - Neural networks

Regulatory Control

- Objective - maintain process performance at a certain level or within a given tolerance band of that level
  - Appropriate when performance relates to a quality measure
- Performance measure is sometimes computed based on several output variables
  - Performance measure is called the *Index of performance* (IP)
- Problem with regulatory control is that an error must exist in order to initiate control action

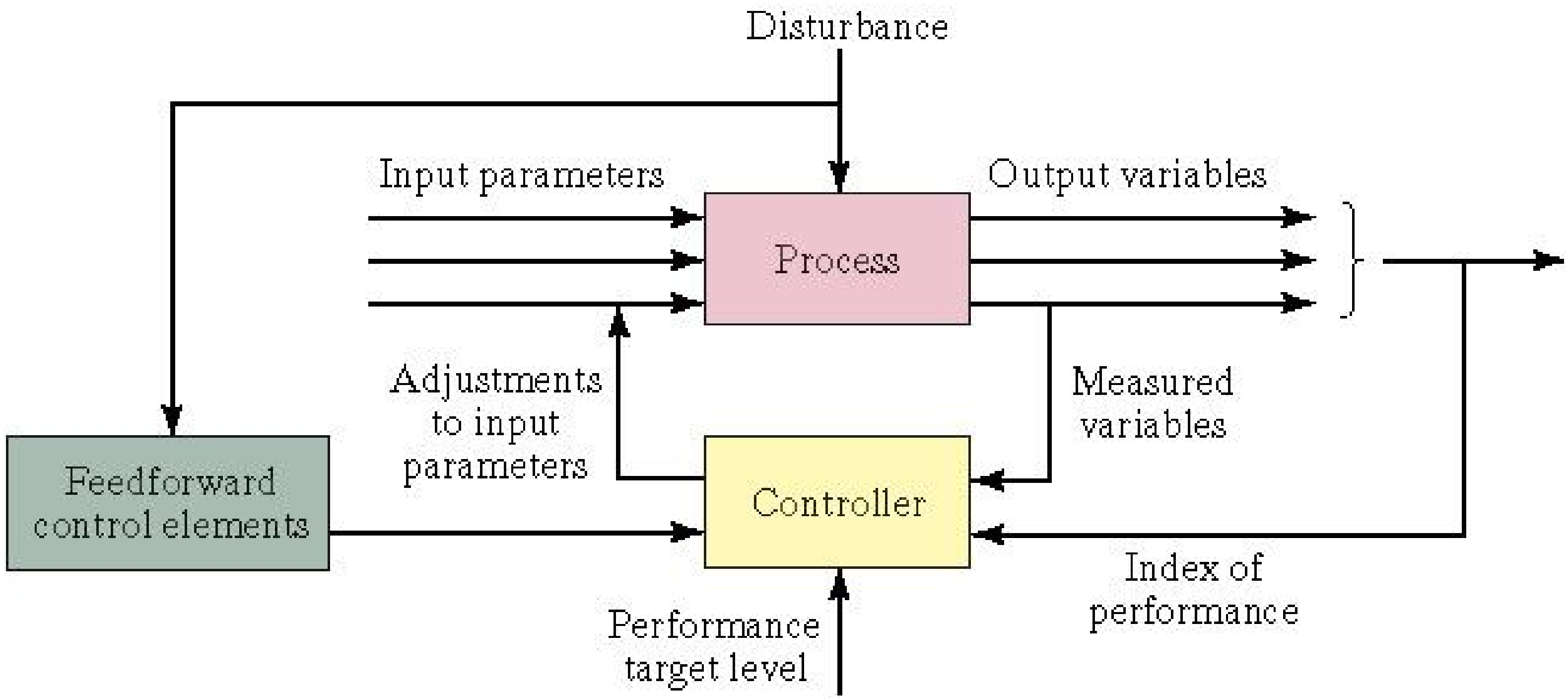
Regulatory Control



Feedforward Control

- Objective - anticipate the effect of disturbances that will upset the process by sensing and compensating for them before they affect the process
- Mathematical model captures the effect of the disturbance on the process
- Complete compensation for the disturbance is difficult due to variations, imperfections in the mathematical model and imperfections in the control actions
  - Usually combined with regulatory control
- Regulatory control and feedforward control

Feedforward Control
Combined with Feedback Control



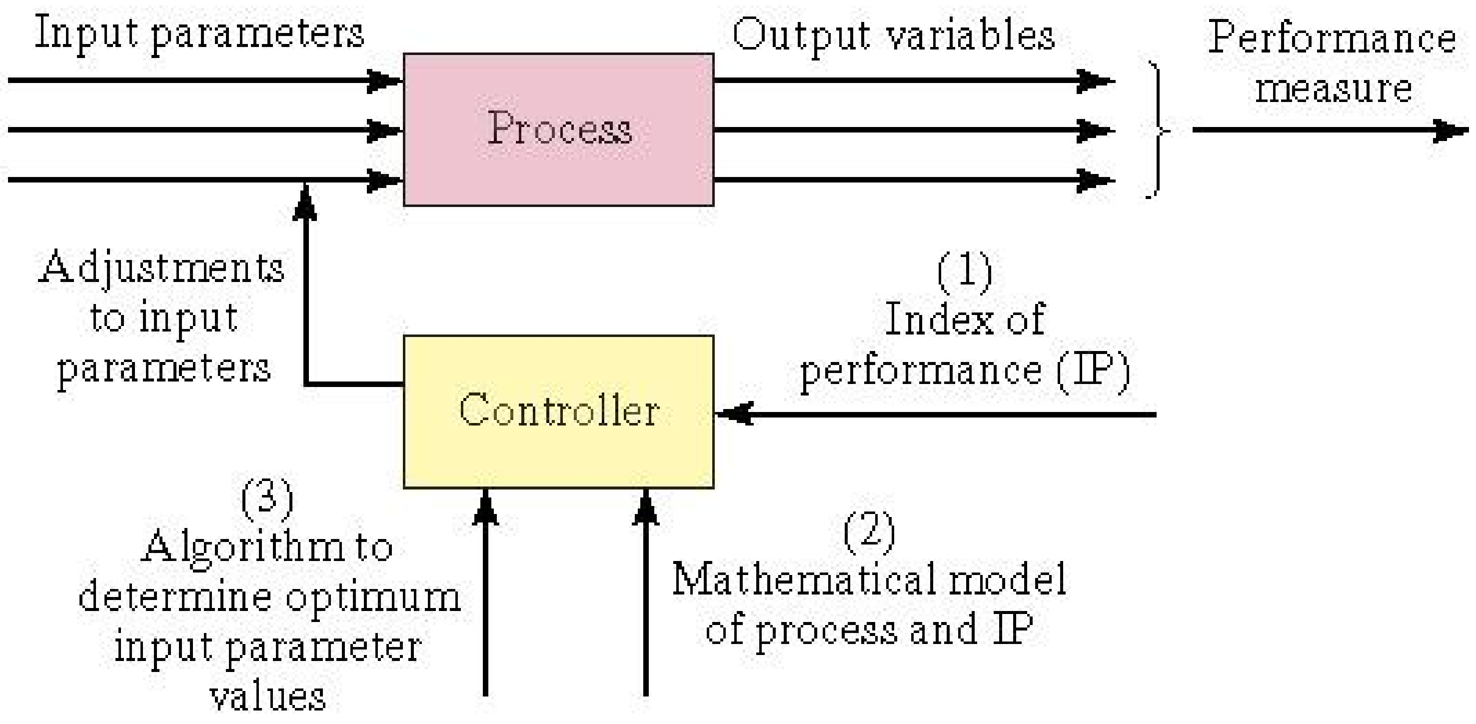
Steady-State Optimization

Class of optimization techniques in which the process exhibits the following characteristics:

1. Well-defined index of performance (IP)
2. Known relationship between process variables and IP
3. System parameter values that optimize IP can be determined mathematically

- Open-loop system
- Optimization techniques include differential calculus, mathematical programming, etc.

Steady State (Open-Loop) Optimal Control



Adaptive Control

- Because steady-state optimization is open-loop, it cannot compensate for disturbances
- Adaptive control is a self-correcting form of optimal control that includes feedback control
  - Measures the relevant process variables during operation (feedback control)
  - Uses a control algorithm that attempts to optimize some index of performance (optimal control)

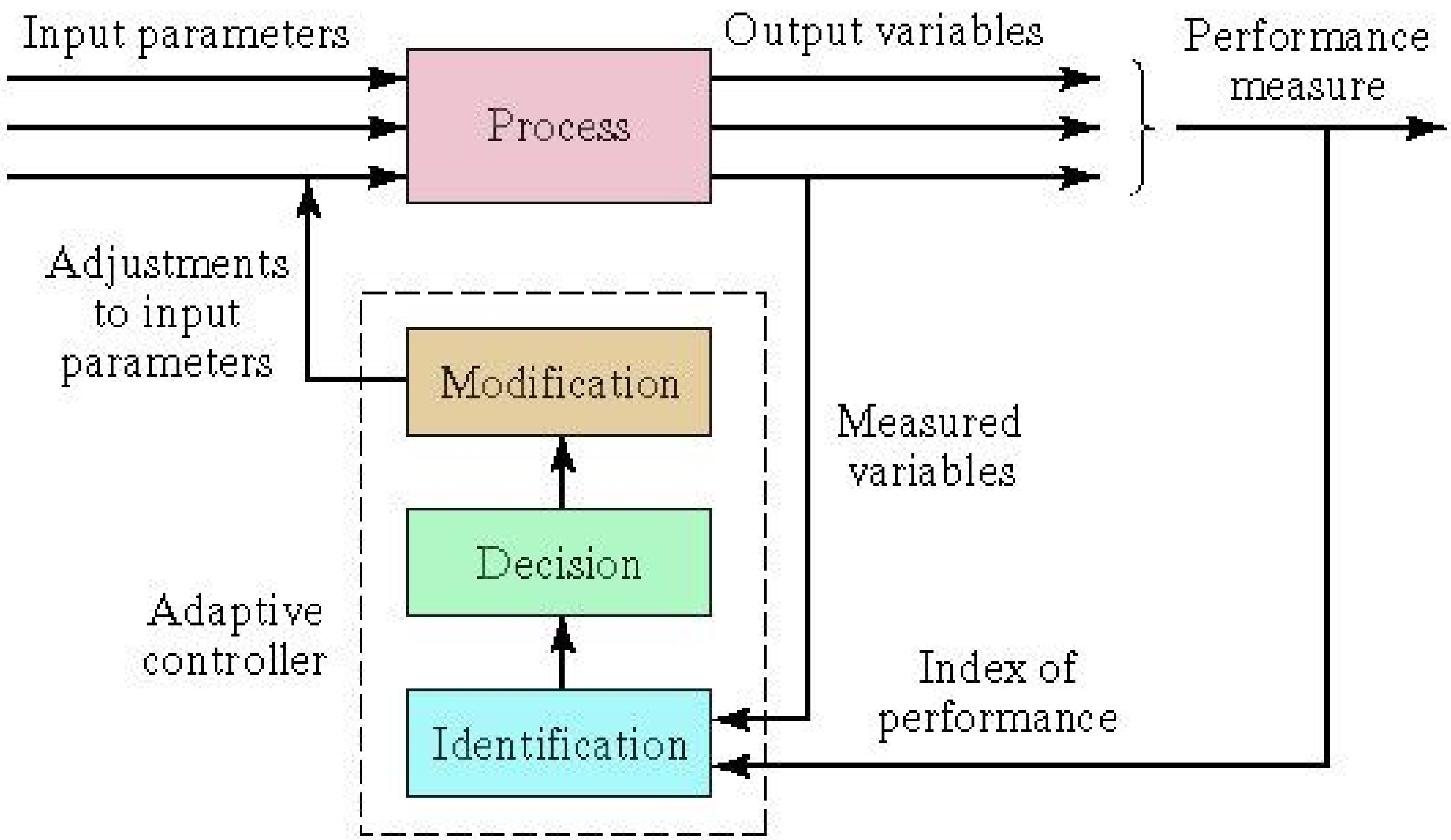
Adaptive Control Operates in a Time-Varying Environment

- The environment changes over time and the changes have a potential effect on system performance
  - Example: Supersonic aircraft operates differently in subsonic flight than in supersonic flight
- If the control algorithm is fixed, the system may perform quite differently in one environment than in another
- An adaptive control system is designed to compensate for its changing environment by altering some aspect of its control algorithm to achieve optimal performance

Three Functions in Adaptive Control

1. Identification function – current value of IP is determined based on measurements of process variables
2. Decision function – decide what changes should be made to improve system performance
   - Change one or more input parameters
   - Alter some internal function of the controller
3. Modification function – implement the decision function
   - Concerned with physical changes (hardware rather than software)

Adaptive Control System



On-Line Search Strategies

- Special class of adaptive control in which the decision function cannot be sufficiently defined
  - Relationship between input parameters and IP is not known, or not known well enough to implement the previous form of adaptive control
- Instead, experiments are performed on the process
  - Small systematic changes are made in input parameters to observe effects
- Based on observed effects, larger changes are made to drive the system toward optimal

Discrete Control Systems

- Process parameters and variables are discrete
- Process parameters and variables are changed at discrete moments in time
- The changes are defined in advance by the program of instructions
- The changes are executed for either of two reasons:
  1. The state of the system has changed (event-driven changes)
  2. A certain amount of time has elapsed (time driven changes)

Event-Driven Changes

- Executed by the controller in response to some event that has altered the state of the system
- Examples:
  - A robot loads a workpart into a fixture, and the part is sensed by a limit switch in the fixture
  - The diminishing level of plastic in the hopper of an injection molding machine triggers a low-level switch, which opens a valve to start the flow of more plastic into the hopper
  - Counting parts moving along a conveyor past an optical sensor

Time-Driven Events

- Executed by the controller either at a specific point in time or after a certain time lapse
- Examples:
  - The factory “shop clock” sounds a bell at specific times to indicate start of shift, break start and stop times, and end of shift
  - Heat treating operations must be carried out for a certain length of time
  - In a washing machine, the agitation cycle is set to operate for a certain length of time
    - By contrast, filling the tub is event-driven

Two Types of Discrete Control

1. Combinational logic control – controls the execution of event-driven changes
    - Also known as logic control
    - Output at any moment depends on the values of the inputs
    - Parameters and variables = 0 or 1 (OFF or ON)
2. Sequential control – controls the execution of time-driven changes
    - Uses internal timing devices to determine when to initiate changes in output variables

Computer Process Control

- Origins in the 1950s in the process industries
  - Mainframe computers – slow, expensive, unreliable
  - Set point control
  - Direct digital control (DDC) system installed 1962
- Minicomputer introduced in late 1960s, microcomputer introduced in early 1970s
- Programmable logic controllers introduced early 1970s for discrete process control
- Distributed control starting around 1975
- PCs for process control early 1990s

Two Basic Requirements for Real-Time Process Control

1. Process-initiated interrupts
   - Controller must respond to incoming signals from the process (event-driven changes)
   - Depending on relative priority, controller may have to interrupt current program to respond
2. Timer-initiated actions
   - Controller must be able to execute certain actions at specified points in time (time-driven changes)
   - Examples: (1) scanning sensor values, (2) turning switches on and off, (3) re-computing optimal parameter values

Other Computer Control Requirements

3. Computer commands to process
   - To drive process actuators
4. System- and program-initiated events
   - System initiated events - communications between computer and peripherals
   - Program initiated events - non-process-related actions, such as printing reports
5. Operator-initiated events – to accept input from personnel
   - Example: emergency stop

Capabilities of Computer Control

- Polling (data sampling)
- Interlocks
- Interrupt system
- Exception handling

Polling (Data Sampling)

Periodic sampling of data to indicate status of process

- Issues:
  1. Polling frequency – reciprocal of time interval between data samples
  2. Polling order – sequence in which data collection points are sampled
  3. Polling format – alternative sampling procedures:
     - All sensors polled every cycle
     - Update only data that has changed this cycle
     - High-level and low-level scanning

Interlocks

Safeguard mechanisms for coordinating the activities of two or more devices and preventing one device from interfering with the other(s)

1. Input interlocks – signal from an external device sent to the controller; possible functions:
   - Proceed to execute work cycle program
   - Interrupt execution of work cycle program
2. Output interlocks – signal sent from controller to external device

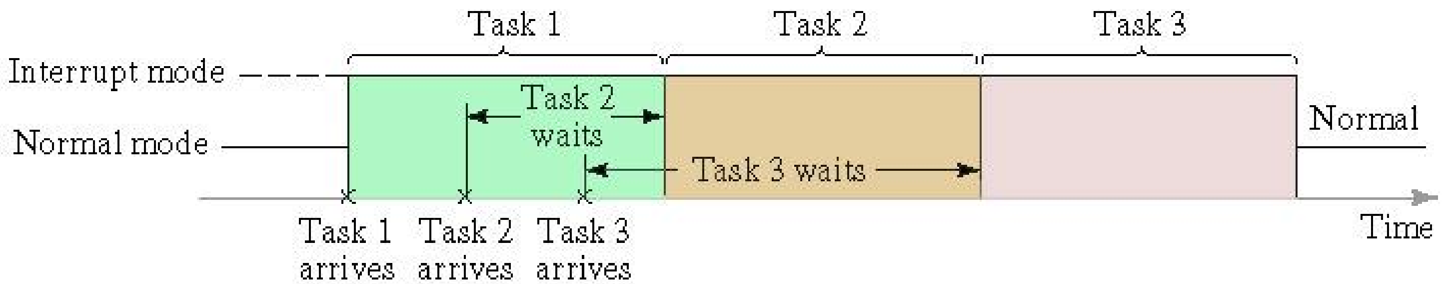
Interrupt System

Computer control feature that permits the execution of the current program to be suspended in order to execute another program in response to an incoming signal indicating a higher priority event

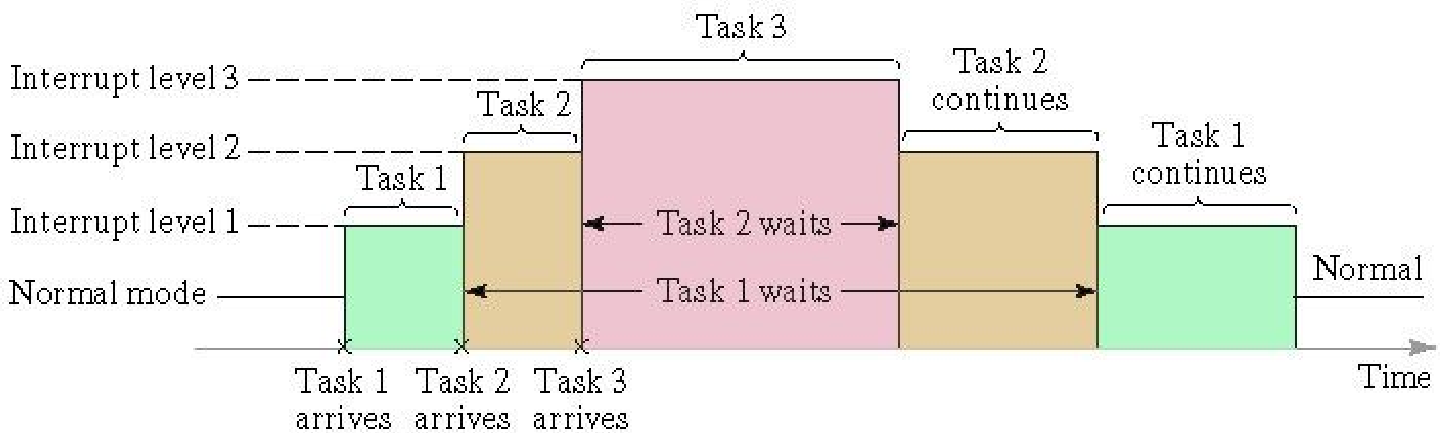
- Internal interrupt – generated by the computer itself
  - Examples: timer-initiated events, polling, system- and program initiated interrupts
- External interrupts – generated external to the computer
  - Examples: process-initiated interrupts, operator

Interrupt Systems:
(a) Single-Level and (b) Multilevel

(a)



(b)



Exception Handling

An exception is an event that is outside the normal or desired operation of the process control system

- Examples of exceptions:
  - Product quality problem
  - Process variable outside normal operating range
  - Shortage of raw materials
  - Hazardous conditions, e.g., fire
  - Controller malfunction
- Exception handling is a form of error detection and recovery

Forms of Computer Process Control

1. Computer process monitoring
2. Direct digital control (DDC)
3. Numerical control and robotics
4. Programmable logic control
5. Supervisory control
6. Distributed control systems and personal computers

Computer Process Monitoring

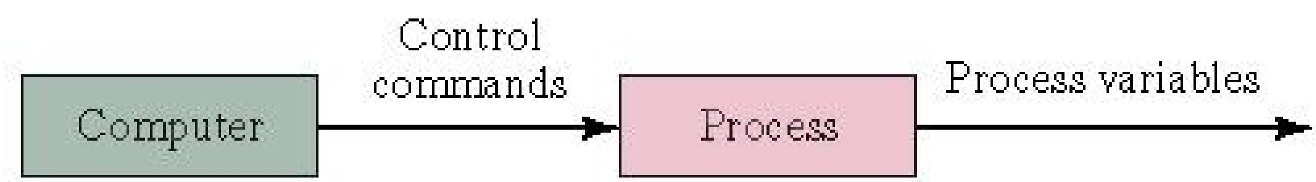
Computer observes process and associated equipment, collects and records data from the operation

- The computer does not directly control the process
- Types of data collected:
  - Process data – input parameters and output variables
  - Equipment data – machine utilization, tool change scheduling, diagnosis of malfunctions
  - Product data – to satisfy government requirements, e.g., pharmaceutical and medical

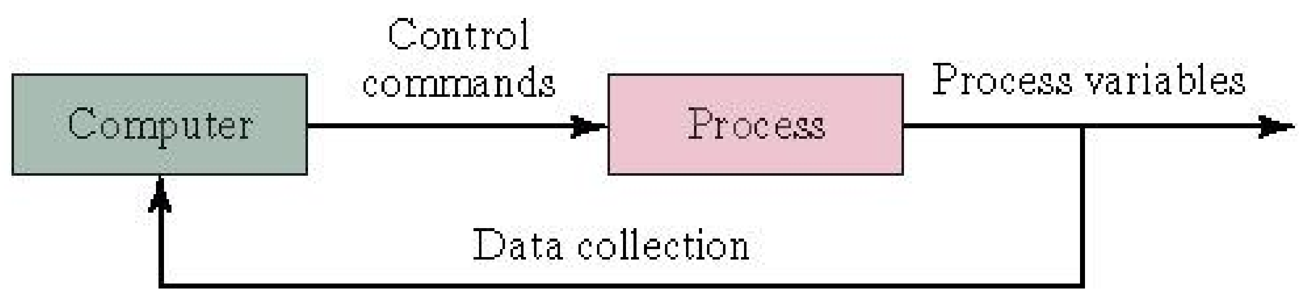
(a) Process Monitoring, (b) Open-Loop Control, and (c) Closed-Loop Control

(a)

(b)



(c)

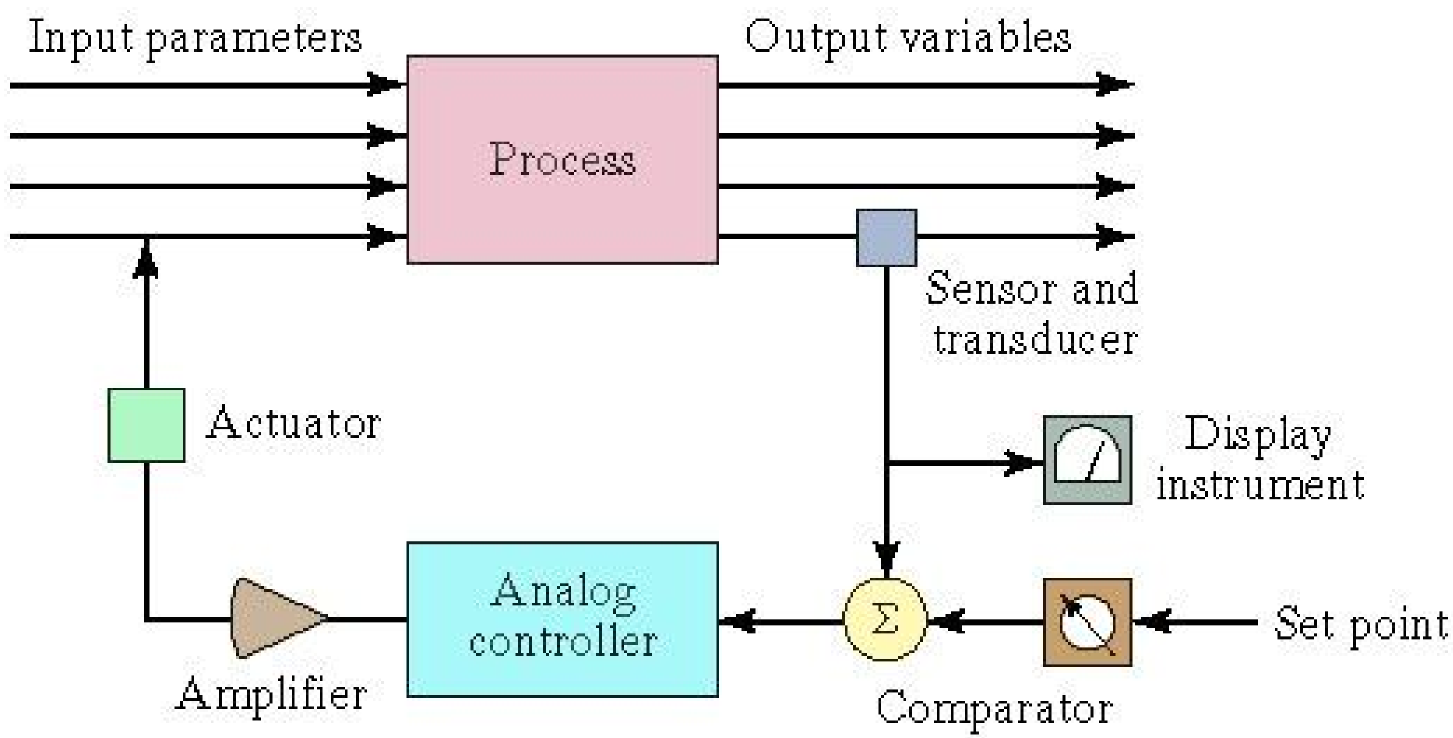


Direct Digital Control (DDC)

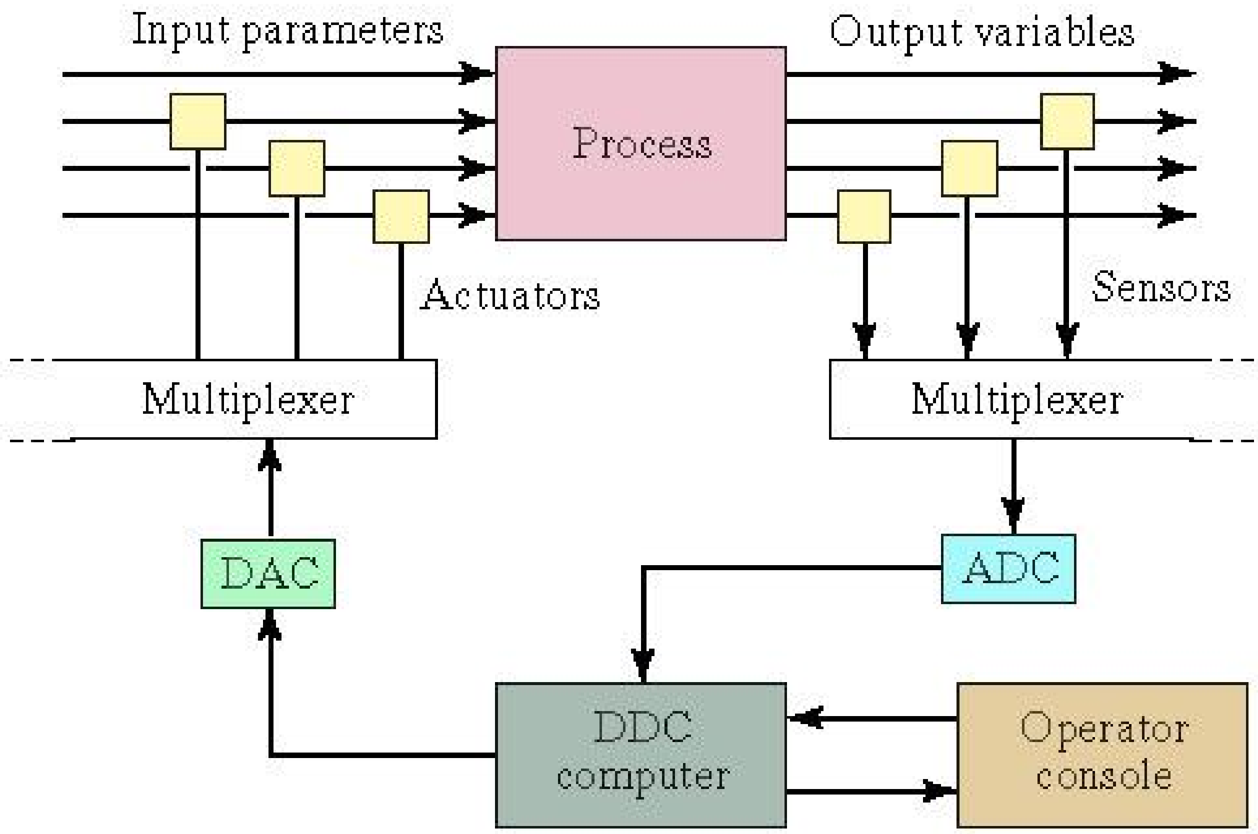
Form of computer process control in which certain components in a conventional analog control system are replaced by the digital computer

- Circa: 1960s using mainframes
- Applications: process industries
- Accomplished on a time-shared, sampled-data basis rather than continuously by dedicated components
  - Components remaining in DDC: sensors and actuators
  - Components replaced in DDC: analog controllers, recording and display instruments, set point dials

A Typical Analog Control Loop



Components of a Direct Digital Control System



DDC (continued)

- Originally seen as a more efficient means of performing the same functions as analog control
- Additional opportunities became apparent in DDC:
  - More control options than traditional analog control (PID control), e.g., combining discrete and continuous control
  - Integration and optimization of multiple loops
  - Editing of control programs

Numerical Control and Robotics

- Computer numerical control (CNC) – computer directs a machine tool through a sequence of processing steps defined by a program of instructions
  - Distinctive feature of NC – control of the position of a tool relative to the object being processed
  - Computations required to determine tool trajectory
- Industrial robotics – manipulator joints are controlled to move and orient end-of-arm through a sequence of positions in the work cycle

Programmable Logic Controller (PLC)

Microprocessor-based controller that executes a program of instructions to implement logic, sequencing, counting, and arithmetic functions to control industrial machines and processes

- Introduced around 1970 to replace electromechanical relay controllers in discrete product manufacturing
- Today's PLCs perform both discrete and continuous control in both process industries and discrete product industries

Supervisory Control

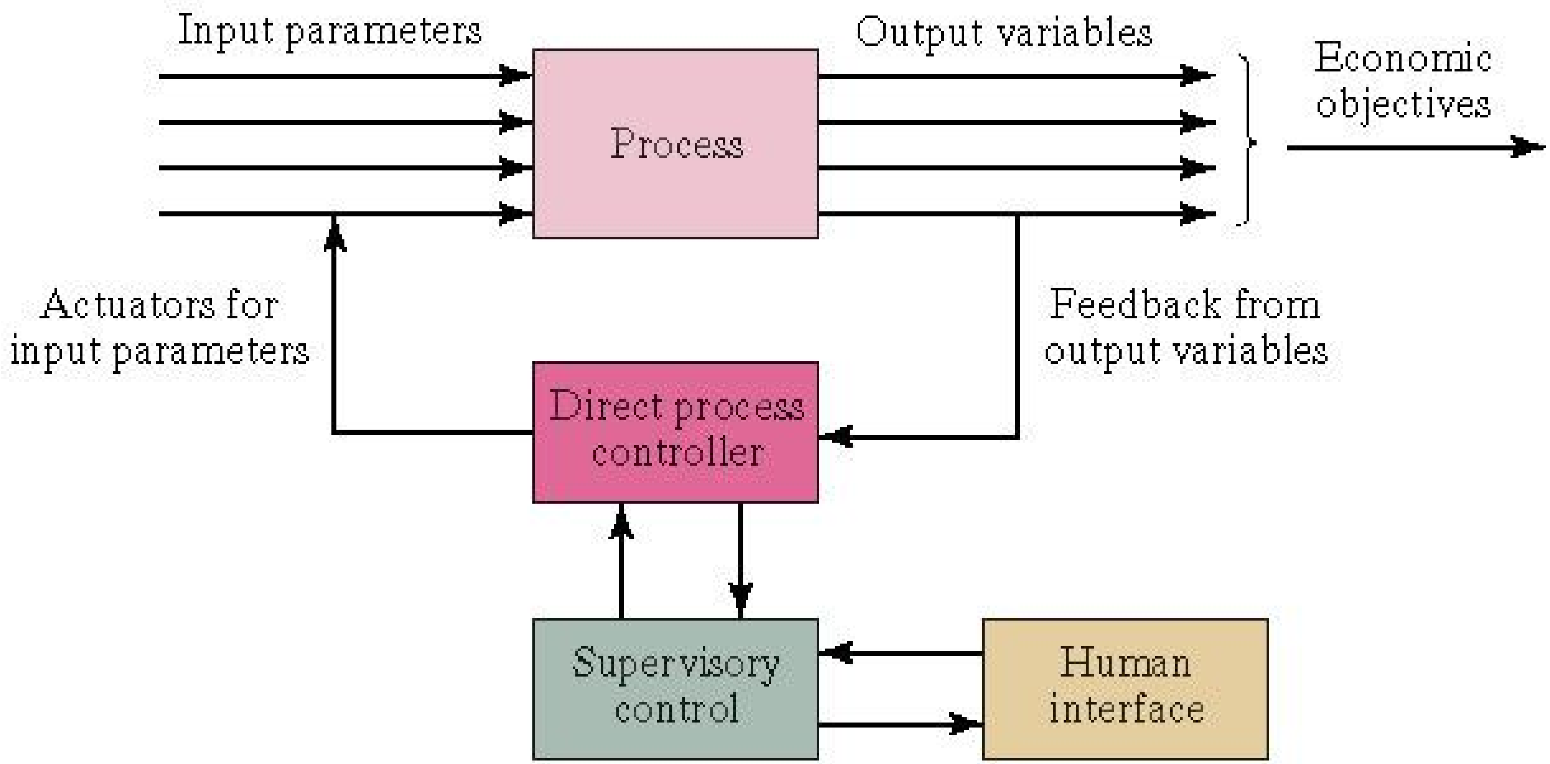
In the process industries, supervisory control denotes a control system that manages the activities of a number of integrated unit operations to achieve certain economic objectives

In discrete manufacturing, supervisory control is the control system that directs and coordinates the activities of several interacting pieces of equipment in a manufacturing system

- Functions: efficient scheduling of production, tracking tool lives, optimize operating parameters

- Most closely associated with the process industries

Supervisory Control Superimposed on Process Level Control System

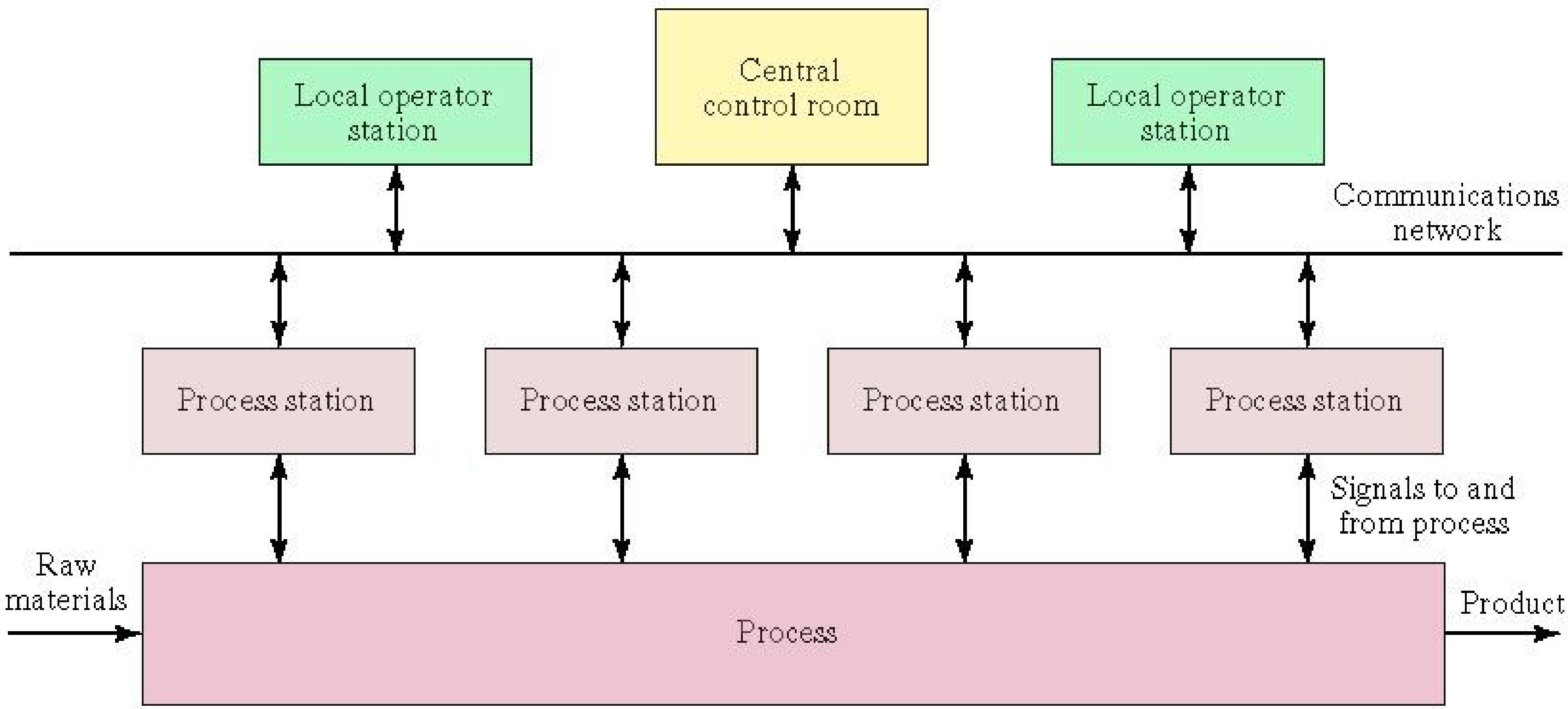


Distributed Control Systems (DCS)

Multiple microcomputers connected together to share and distribute the process control workload

- Features:
  - Multiple process control stations to control individual loops and devices
  - Central control room where supervisory control is accomplished
  - Local operator stations for redundancy
  - Communications network (data highway)

Distributed Control System



DCS Advantages

- Can be installed in a very basic configuration, then expanded and enhanced as needed in the future
- Multiple computers facilitate parallel multitasking
- Redundancy due to multiple computers
- Control cabling is reduced compared to central controller configuration
- Networking provides process information throughout the enterprise for more efficient plant and process management

PCs in Process Control

Two categories of personal computer applications in process control:

1. Operator interface – PC is interfaced to one or more PLCs or other devices that directly control the process
   - PC performs certain monitoring and supervisory functions, but does not directly control process
2. Direct control – PC is interfaced directly to the process and controls its operations in real time
   - Traditional thinking is that this is risky

Enablers of PCs for Direct Control

- Widespread familiarity of workers with PCs
- Availability of high performance PCs
  - Cycle speeds of PCs now exceed those of PLCs
- Open architecture philosophy in control system design
  - Hardware and software vendors comply with standards that allow their products to be interoperable
- PC operating systems that facilitate real-time control and networking
- PC industrial grade enclosures

Enterprise-Wide Integration of Factory Data

- Managers have direct access to factory operations
- Planners have most current data on production times and rates for scheduling purposes
- Sales personnel can provide realistic delivery dates to customers, based on current shop loading
- Order trackers can provide current status information to inquiring customers
- QC can access quality issues from previous orders
- Accounting has most recent production cost data
- Production personnel can access product design data to clarify ambiguities

Enterprise-Wide PC-based Distributed Control System