Embedded System Case Studies

COE718: Embedded Systems Design
http://www.ecb.torontomu.ca/~courses/coe718/

Dr. Gul N. Khan
http://www.ecb.torontomu.ca/~gnkhan
Electrical and Computer Engineering
Toronto Metropolitan University

Overview

- CRC cards
- Elevator Control Design
- HP Plotter

Part of Chapter 8 (Section 8.6) and 9 (Section 9.8) of Text by Wolf

Elevator Systems

CRC cards is a well-known method for analyzing a system and developing an architecture.

CRC

- Classes
- Responsibilities
- Collaborators

Elevator Control Classes

Elevator car, Passenger, Floor control, Car control, Car sensors, etc.

Architectural Classes

Car state, Floor control reader, Car control reader, Car control sender, Scheduler.

Elevator Responsibilities and Collaborators

class	responsibilities	collaborators
Elevator car*	Move up and down	Car control, car sensor, car control sender
Car control*	Transmits car requests	Passenger, floor control reader
Car state	Reads current position of car	Scheduler, car sensor

Elevator System

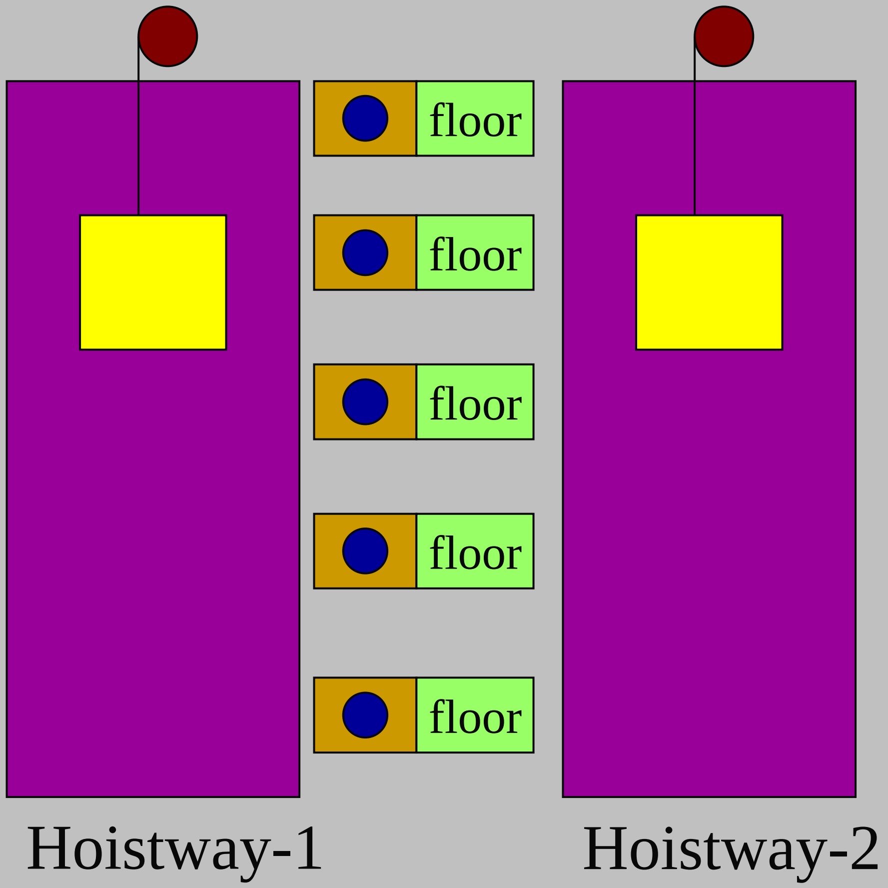
Terminology

Elevator Car

Hoistway

Car control panel

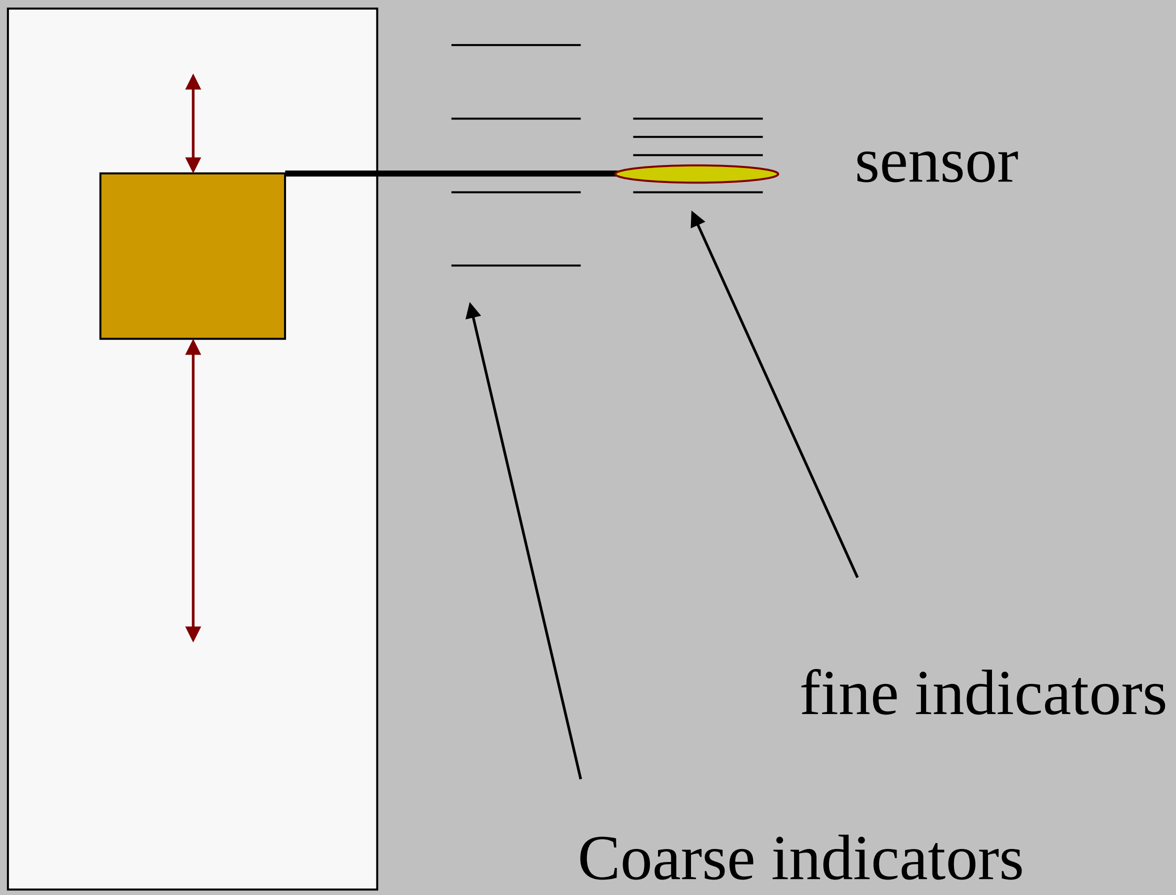
Floor control panel:



Elevator Operation

- **Each floor has control panel, display**
- **Each car has control panel:**
- **Controlled by a single controller**
- **Elevator control has up and down**
  - To stop, disable both
- **Master controller:**
  - reads elevator positions
  - reads requests
  - schedules elevators
  - controls movement
  - controls doors

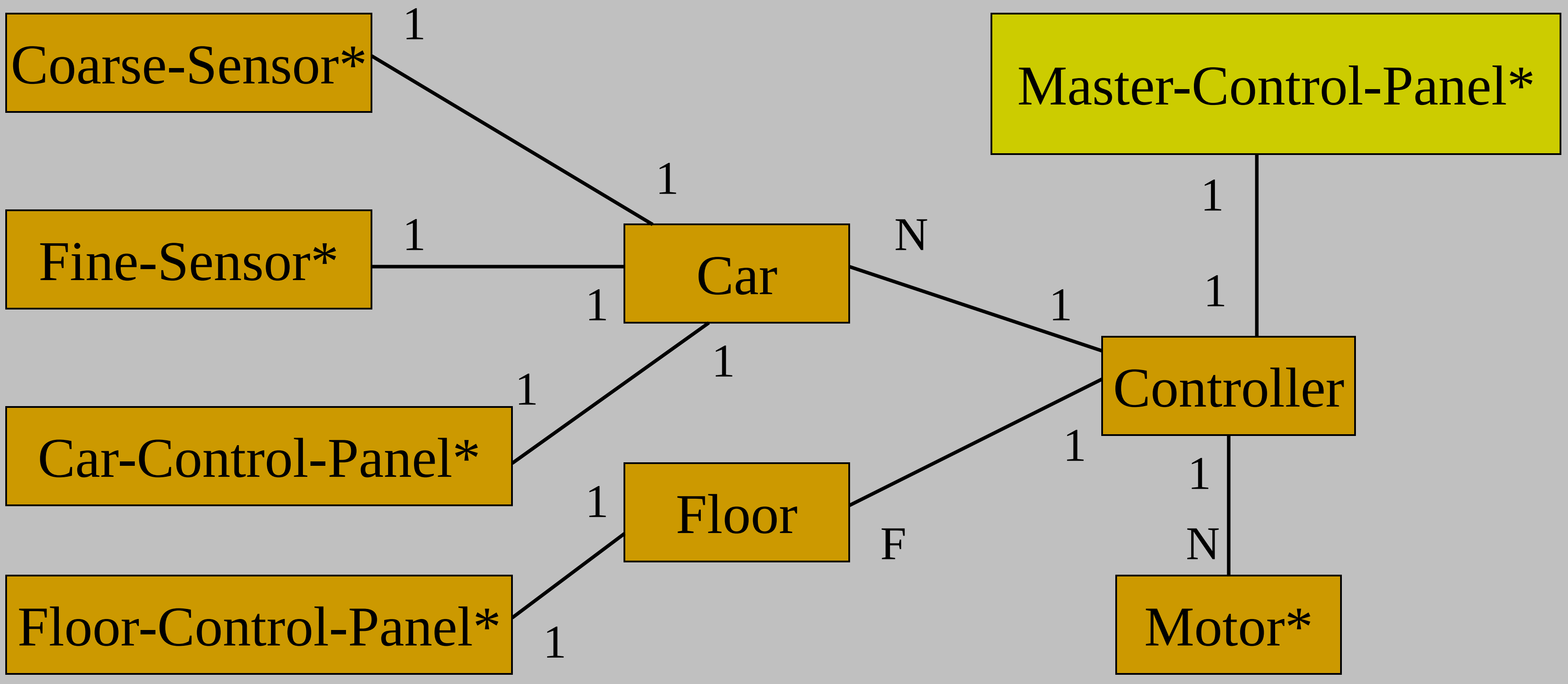
Elevator Position Sensing



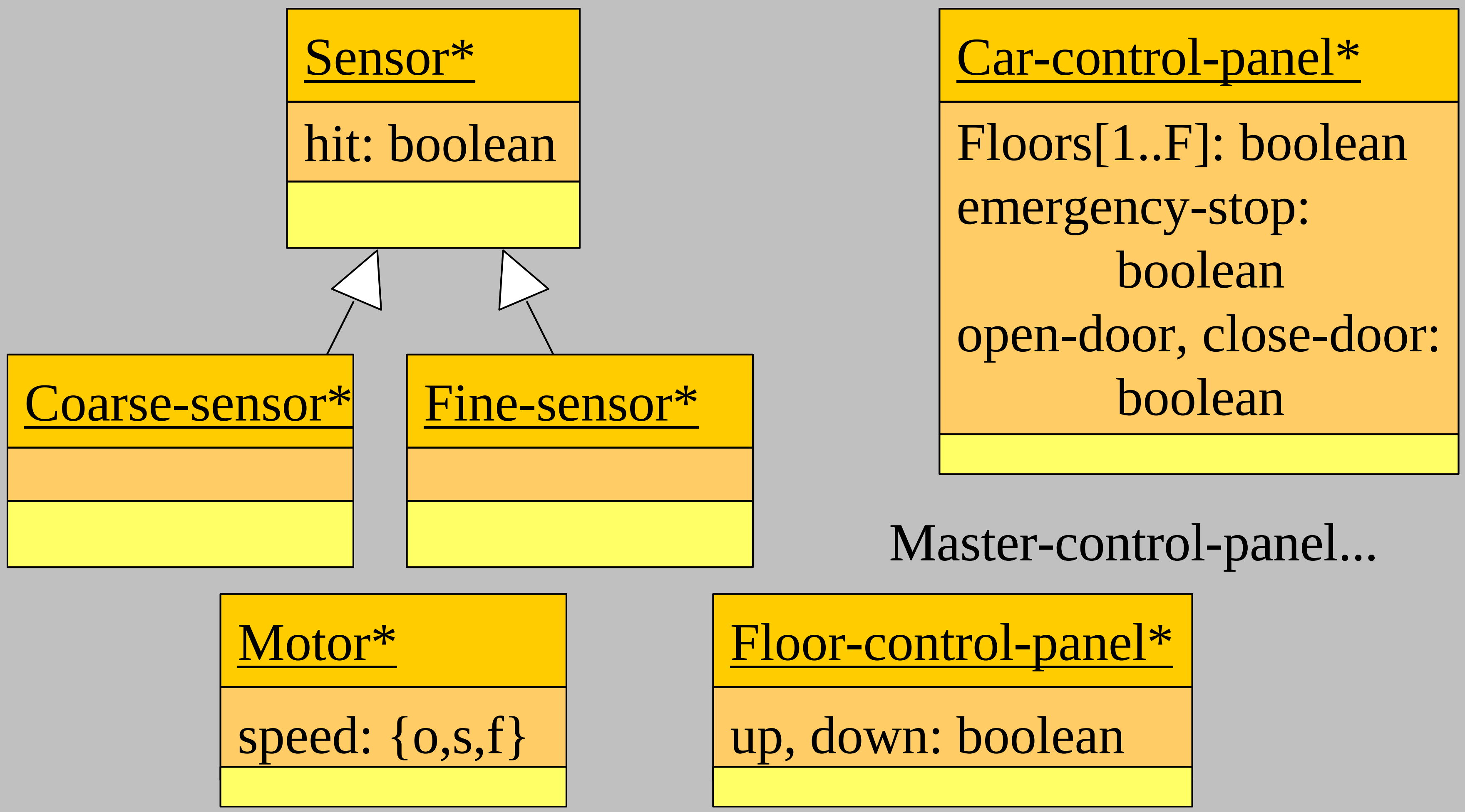
Elevator System Requirements

name	elevator system
inputs	F floor control, N position, N car control, 1 master
outputs	F displays, N motor controllers
functions	responds to requests, operates safely
performance	elevator control is time-critical
manufacturing cost	electronics is small part of total
power	electronics consumes small fraction of total
physical size/weight	cabling is important

Elevator System Classes



Physical Interfaces



Elevator Classes

Controller Class

Controller
car-floor[1..N]: integer emergency-stop[1..N]: integer
scan-cars() scan-floors() scan-master-panel() operate()

Car and Floor classes

Car
request-lights[1..F]: boolean current-floor: integer

Floor
up-light, down-light: boolean

Architecture

Computation and I/O occur at:

- **Floor control panels/displays:**
  - Simple and they have no hard real-time requirements. Take up/down inputs and light the appropriate lights.
  - Watches for button events and sends result to the system controller
  - Use a simple μ-controller for such tasks
- **Elevator Cabs: Cab Controller**
  - read buttons and send events to system controller
  - read sensor inputs & send to system controller, hard real-time.
  - Cab controller must not miss any sensor (fine/course indicator)
- **System controller**

Architecture

System Controller

- Must take inputs from all the Cab controllers, floor control panels.
- It has both hard and soft real-time tasks.
- It must monitor all the moving elevators to make sure that they stop properly.
- Choose which elevator to dispatch to a request.
- Each elevator has a point-to-point connect to system controller.
- Floor control panels/displays can be connected with a single bus network.

System Controller

Take inputs from many sources:

Must control cars to hard real-time deadlines.

User interface via floor control panels and scheduling of elevators are soft real-time deadlines

Testing

Build an elevator simulator using SystemC and/or FPGA

- Simulate multiple elevators
- Simulate real-time control demands.

HP DesignJet Drafting Plotter

Plots up to 36 inches wide at 300 DPI.

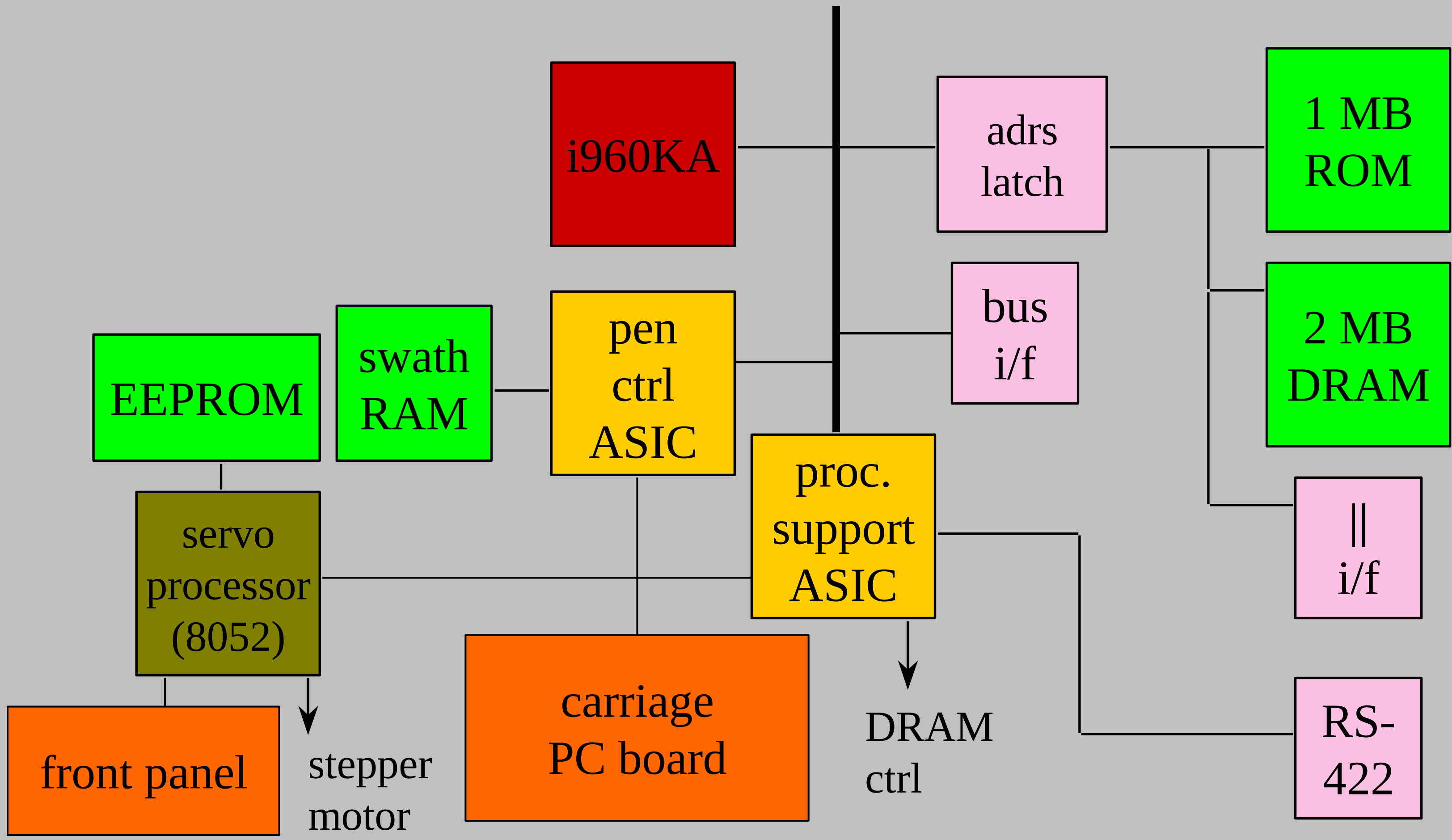
Combines a variety of tasks:

Design Considerations

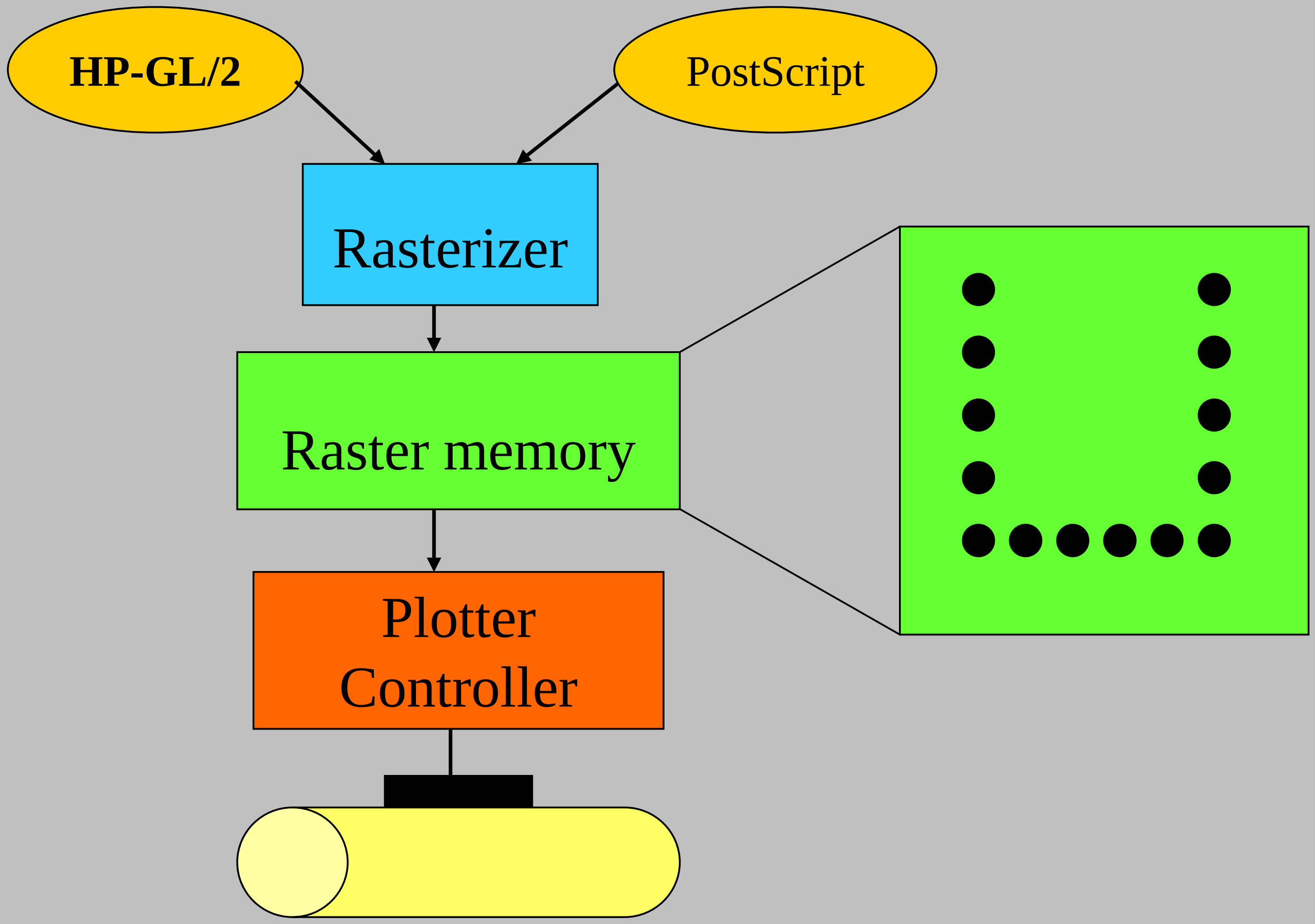
Memory utilization is important

- 36 inches X large X 300 DPI X n bits/pixel is a lot of memory
- Needs clever algorithms to minimize raster memory requirements
- Requires real-time control
- Requires concurrency

HP DesignJet HW Architecture



The Plotting Process

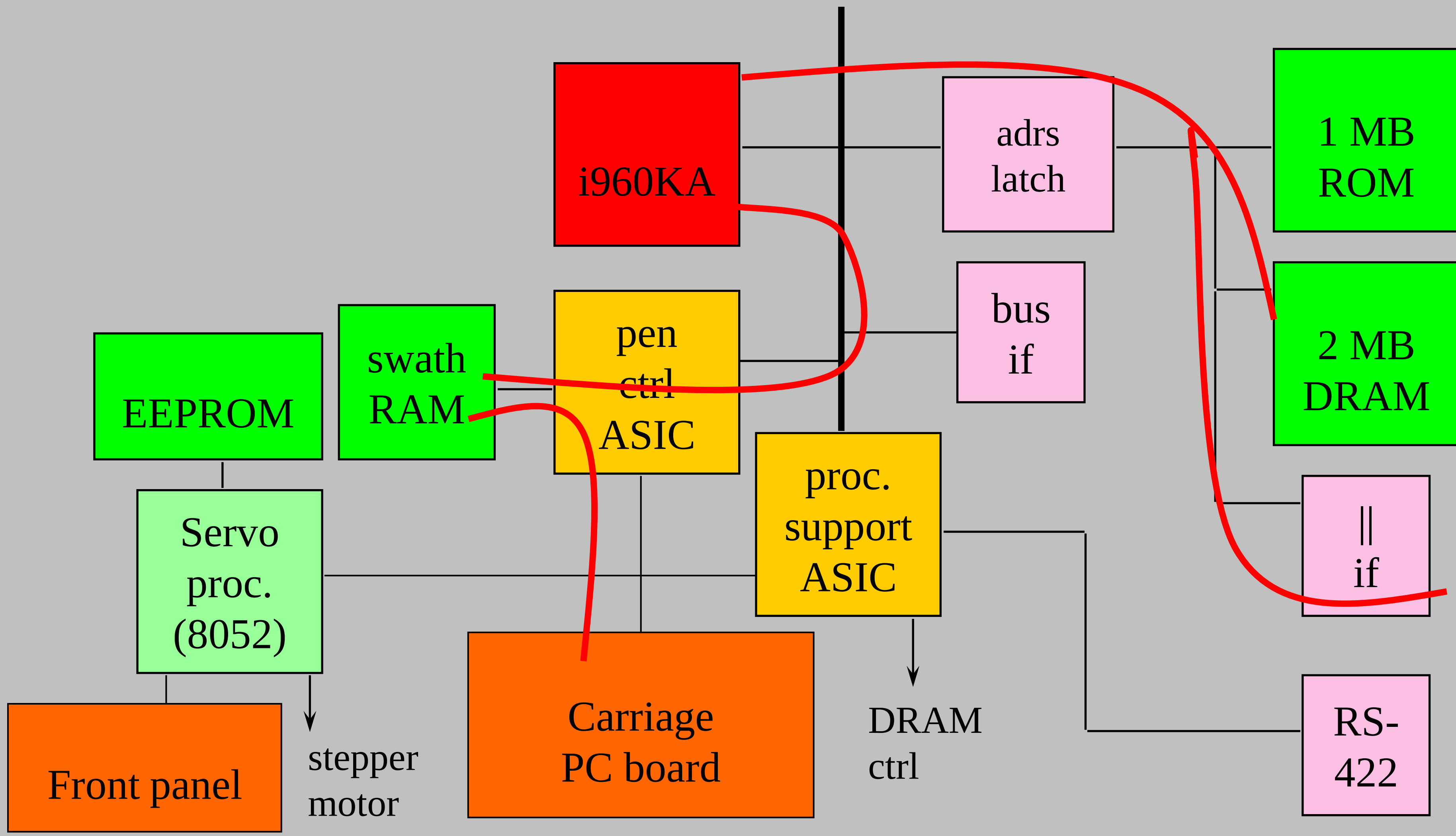


Early Architectural Decisions

- **Chose Intel 80960KA as main processor**
  - Handled parsing, rasterization control, print engine control.
  - Multiplexed bus reduced pin count.
  - Could be upgraded to floating-point if necessary.
- **Used modular I/O to host system.**
- **Did not use disk for local storage.**
- **System components**
- **2 MB RAM (SIMM sockets for more).**
- **Three ASICs:**

- **Servo processing performed by 8052 Microcontroller**

Data Flows



i960KA
adrs latch
1 MB ROM
bus if
2 MB DRAM
EEPROM
swath RAM
pen ctrl ASIC
proc. support ASIC
|| if
Servo proc. (8052)
Carriage PC board
DRAM ctrl
RS-422
Front panel
stepper motor

Rasterization and Operations

Rasterization

- Plot is generated in swaths.
- Pixels are generated in row order by main processor.
- Pixels are fed to pens in column order.
- Pen interface ASIC transforms row order to column order.

Operations

- Servo processor controls stepper motor.
- Carriage processor must write, read pen alignment marks.
- Processor support ASIC provides multiple functions.
- Motion controller decodes position of print carriage and paper.

Pen Interface and Carriage ASICs

Pen Interface ASIC

- Interfaces to i960 bus, swath memory, carriage ASIC.
- Pen interface reads pixels from swath in predetermined pattern using pixel address generator.
- Must support bi-directional printing since head prints both ways.

Carriage ASIC

- Interfaces to processor bus, pen interface ASIC, servo controller.
- Reads timing control registers using the CPU bus.
- Delay registers add correction for pen alignment.

Development Process

Pixel shuffling algorithm for pen interface/carriage ASICs was prototyped in C.

Software Development Environment

- Plotter software could be run on Unix workstation or target platform.

- Used in-house RTOS, HP-GL/2 parser was legacy code
- Rewrote vector/raster converter from assembly language to C to port to i960.
- Front panel developed on PC, tested by user i/f designers/marketing.
- Paper loading designed by mechanical engineers.

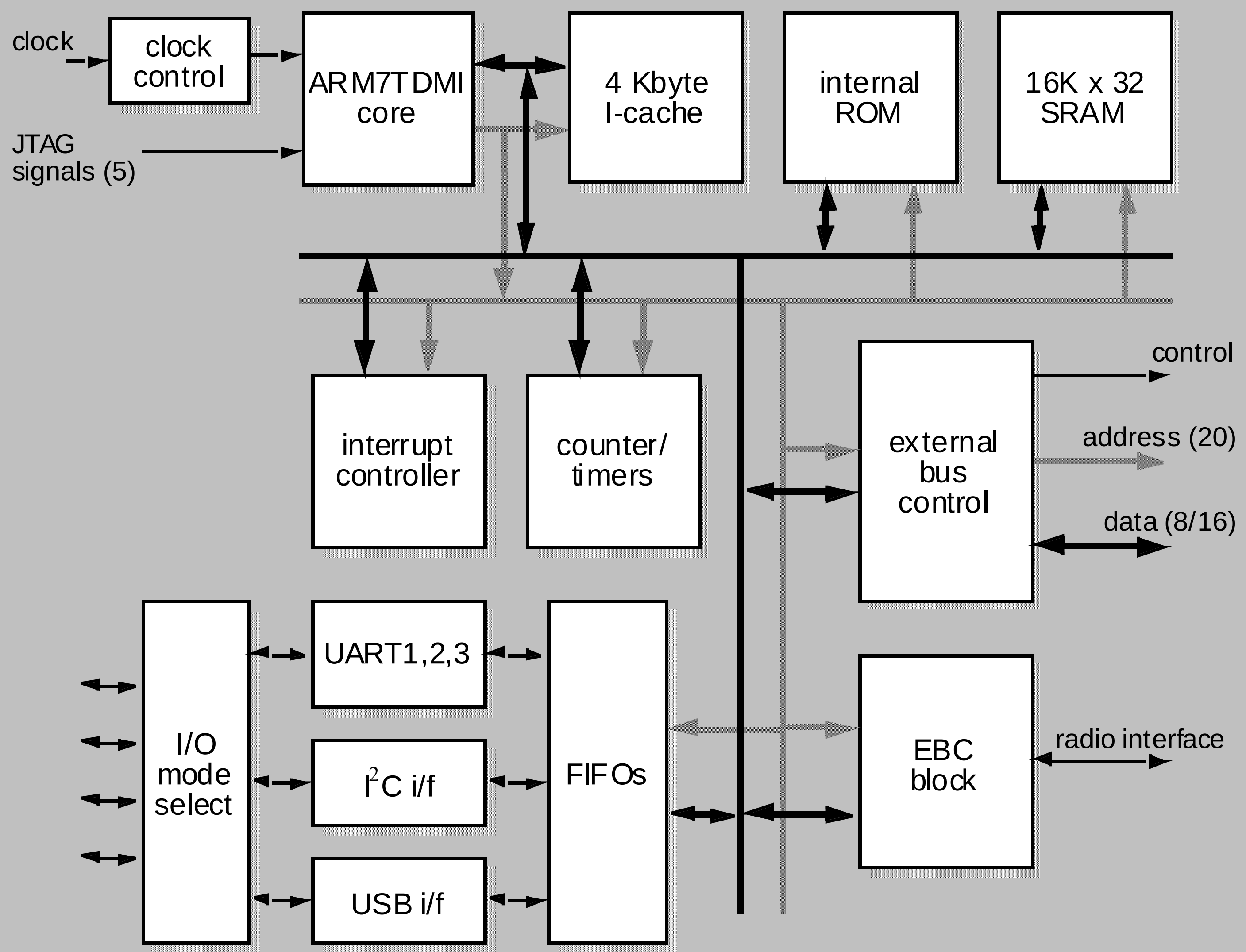
Bluetooth Baseband Controller

Bluetooth is a de-facto standard for wireless data communication for 2.4GHz band. It is developed by a consortium including, Ericsson, Intel, Nokia and Toshiba.

Bluetooth Support

- Short-range communication (10cm to 10m)
- Intends to support laptop to cell phone, printer, fax machines, keyboards, etc.
- Provide a bridge to existing data networks.
- A gross data rate of 1Mbit/s
- Uses frequency hopping scheme and forward error correction.
- Robust communication in a noisy and uncoordinated environment.

VLSI Bluetooth Baseband Controller Organization



Bluetooth Baseband Controller

Uses an ARM7TDMI core with

- 64Kbytes of fast RAM.
- 4Kbyte of instruction cache.

Sharing pin peripheral modules of 3 UARTS & a USB interface

Bluetooth Baseband Controller includes a power optimized hardware block, the Ericsson Bluetooth Core (EBC) that handles all the Link Controller functionality.

- EBC performs all the packet handling functions for point-to-point, multi-slot and point-to-multipoint communications.
- The protocol uses a combination of circuit and packet switching.
- Slots that can be reserved for synch channels (e.g. to support voice transmission).