# System Architecture for Software Peripherals

Siddharth Choudhuri, Tony Givargis

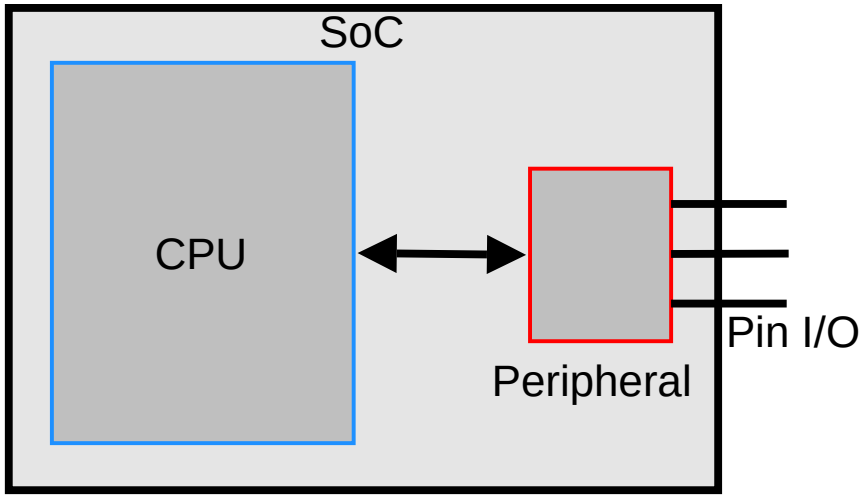
sid@cecs.uci.edu givargis@uci.edu

Center for Embedded Computer Systems
University of California, Irvine, USA

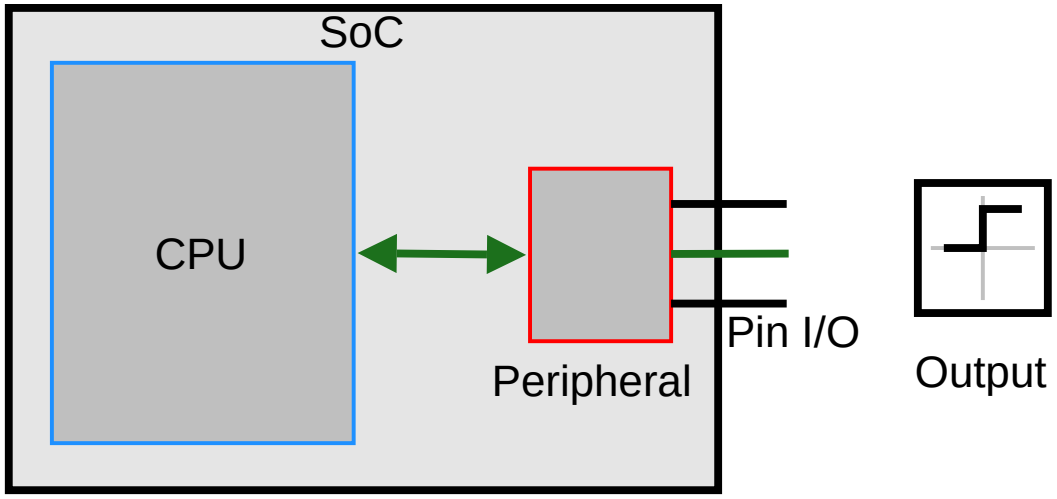
# Outline

- Software Peripherals - Introduction
- Software Peripherals - Advantages
- Related Work
- Proposed System Architecture
- Experimental Setup
- Results

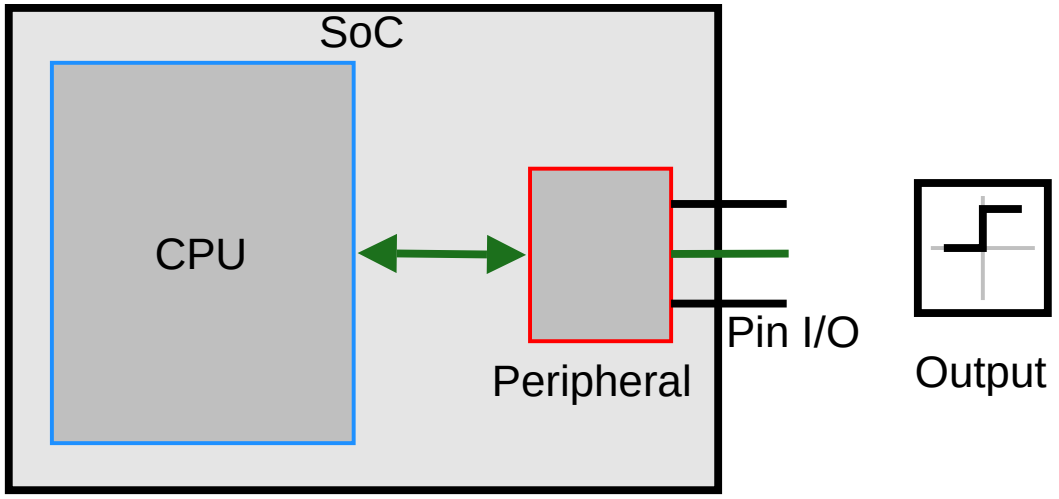
# Peripherals - Traditional View



# Peripherals - Traditional View

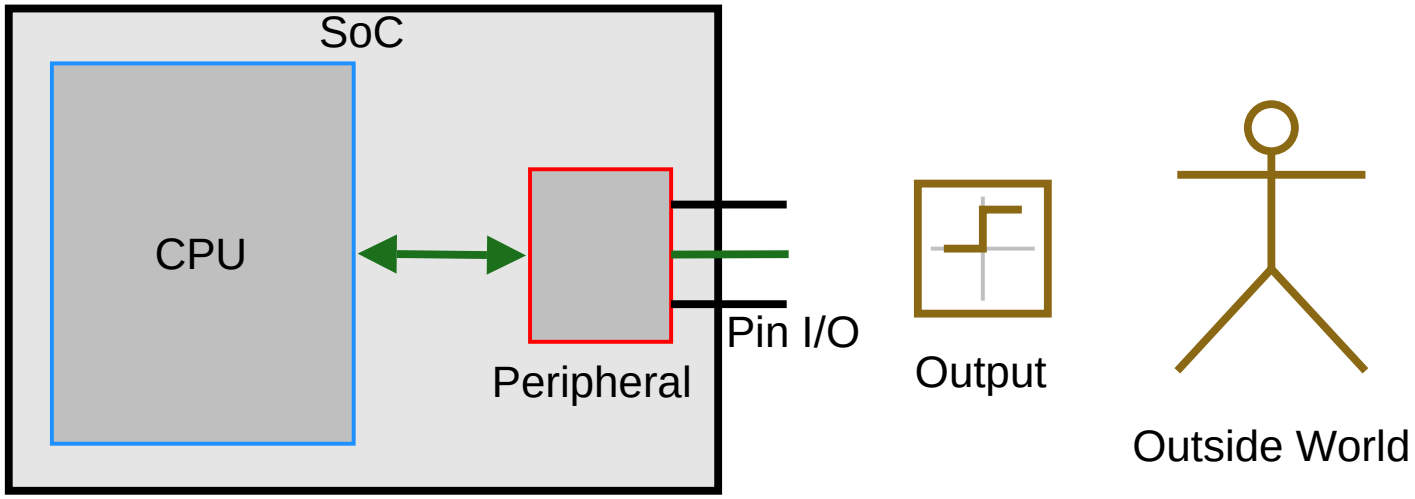


# Peripherals - Traditional View



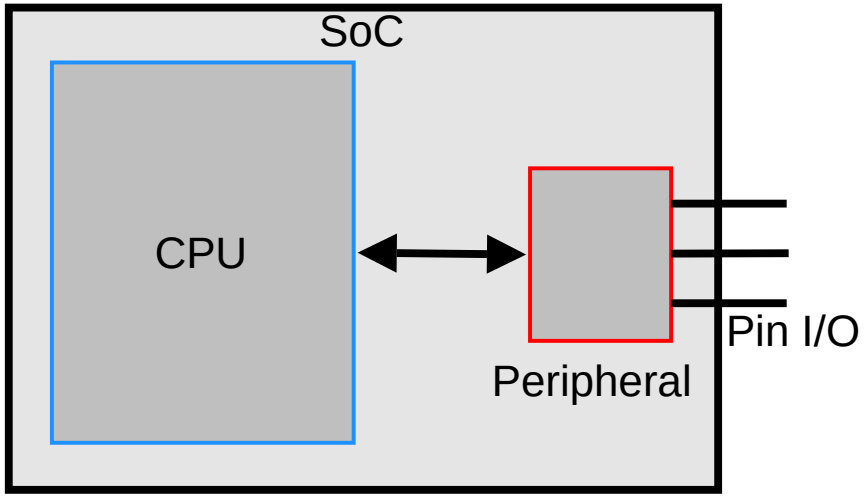

- Periphal is a hardware entity
  - High NRE on redesign
  - Lower flexibility - Hardcoded
  - Cannot be upgarded in field

# Peripherals - Traditional View



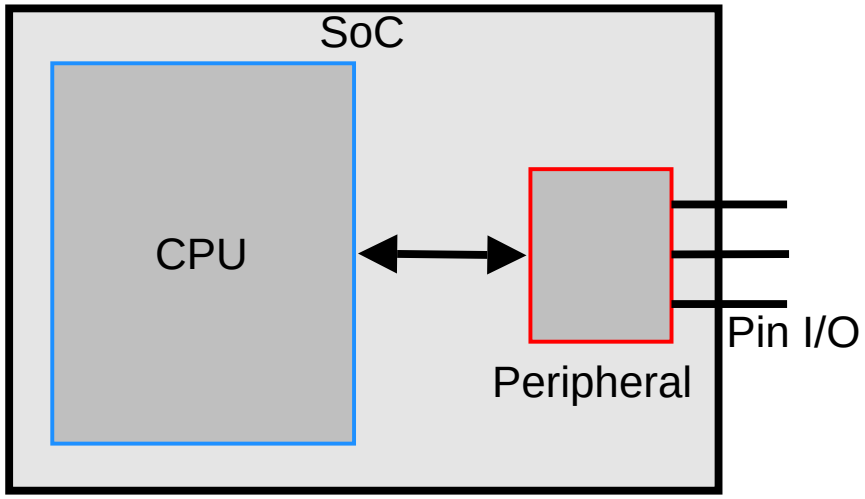

- Periphal is a hardware entity
  - High NRE on redesign
  - Lower flexibility - Hardcoded
  - Cannot be upgarded in field
- Interface view - outside world
  - Output waveform - *rate* and *nature*

# Software Peripherals

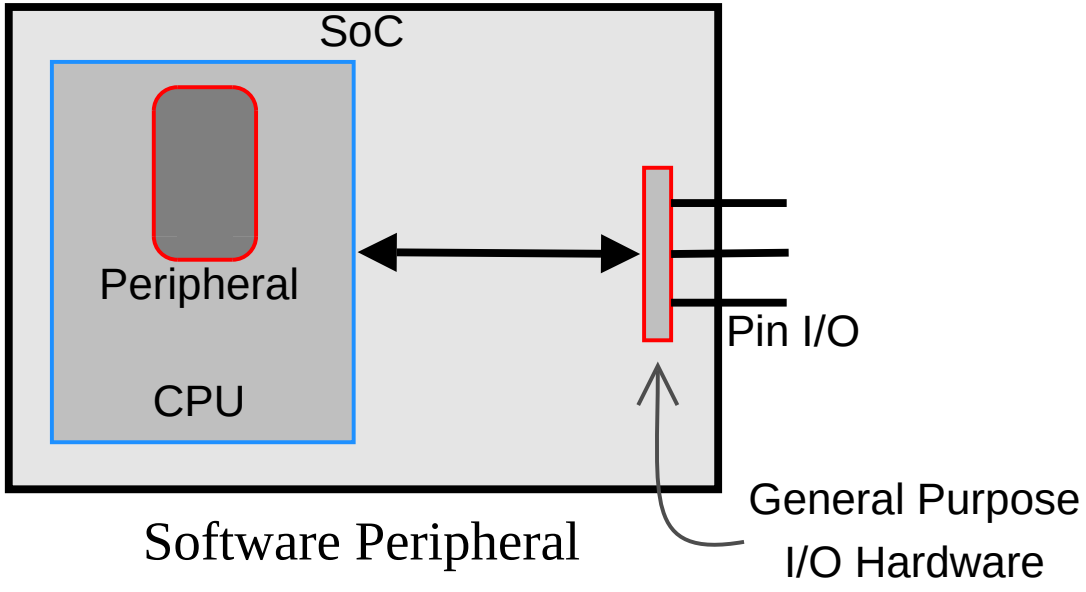



Traditional View

# Software Peripherals

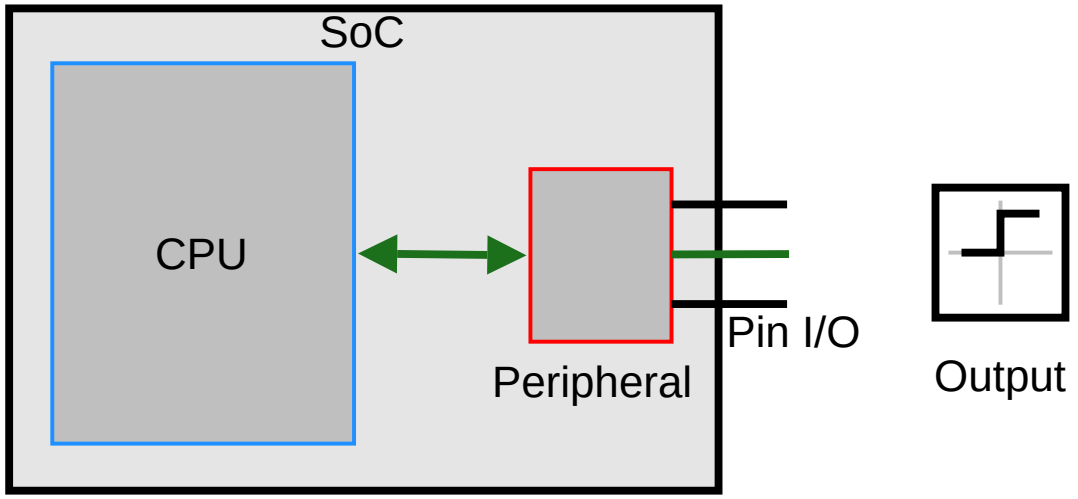



Traditional View

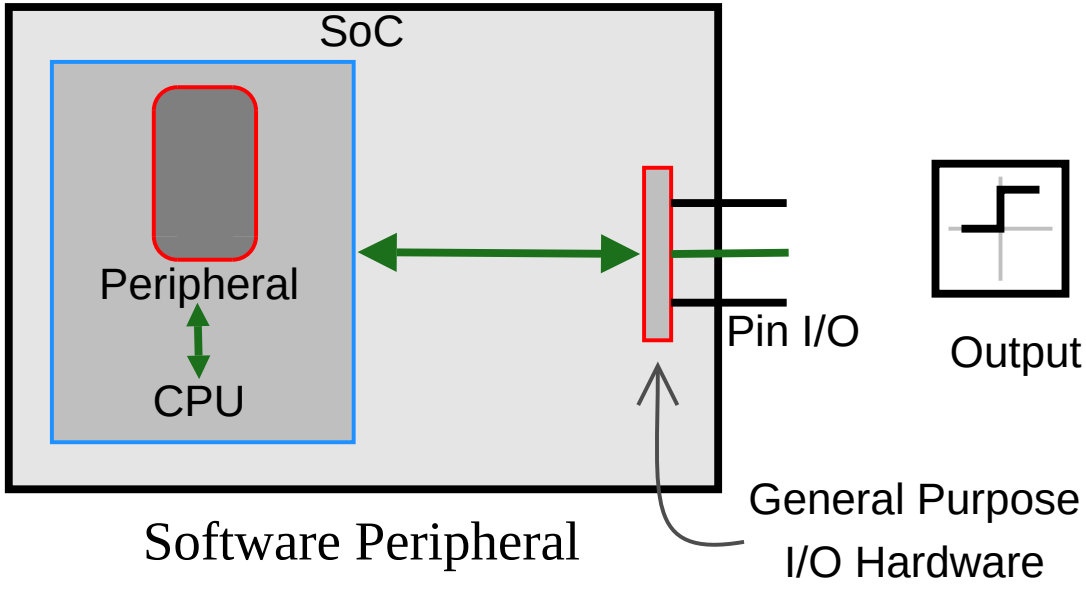



Software Peripheral

# Software Peripherals

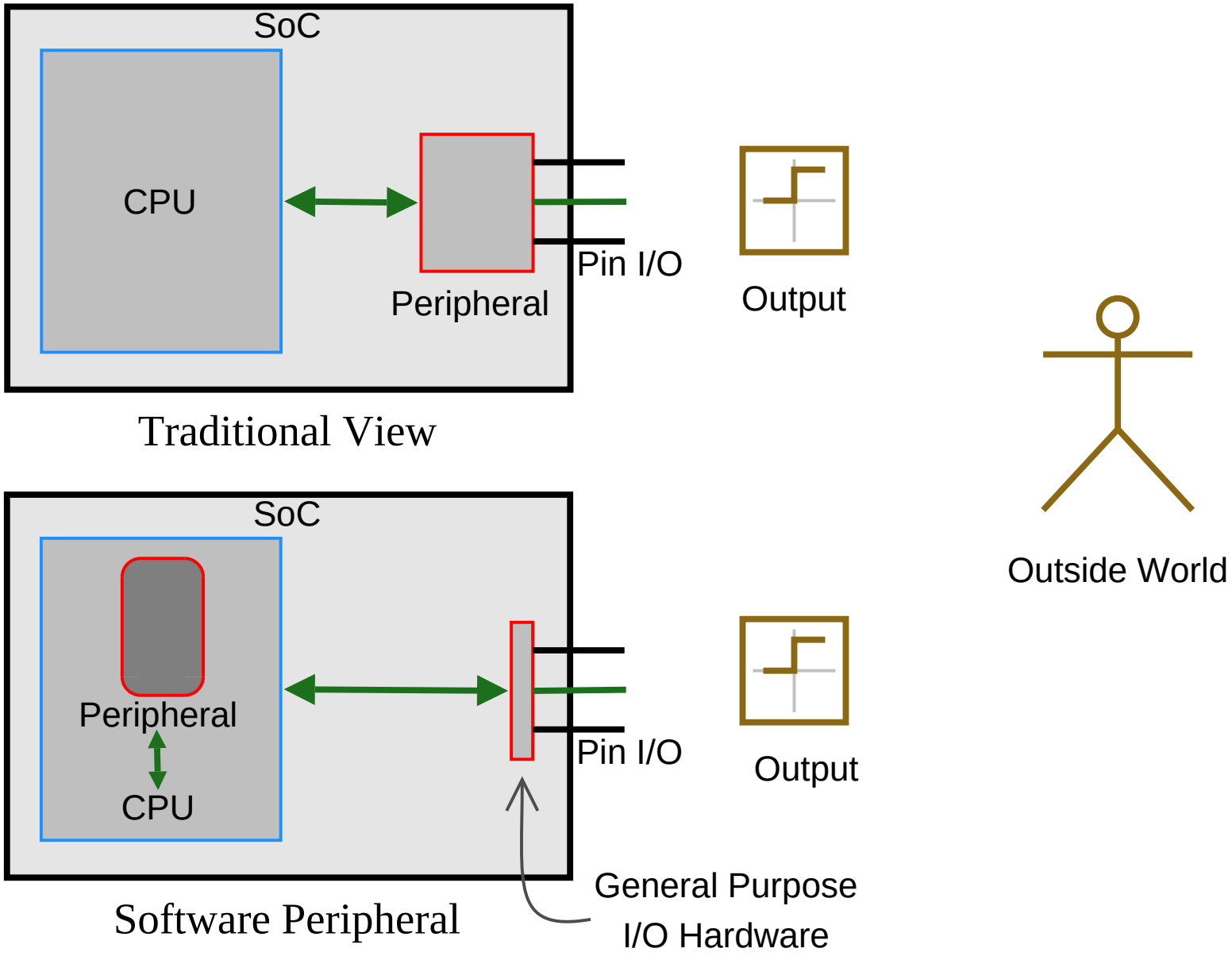



Traditional View




Software Peripheral

# Software Peripherals



# Advantages of Software Peripherals

- Design reusability

# Advantages of Software Peripherals

- Design reusability
- Easier upgrades

# Advantages of Software Peripherals

- Design reusability
- Easier upgrades
- Reduced manufacturing cost

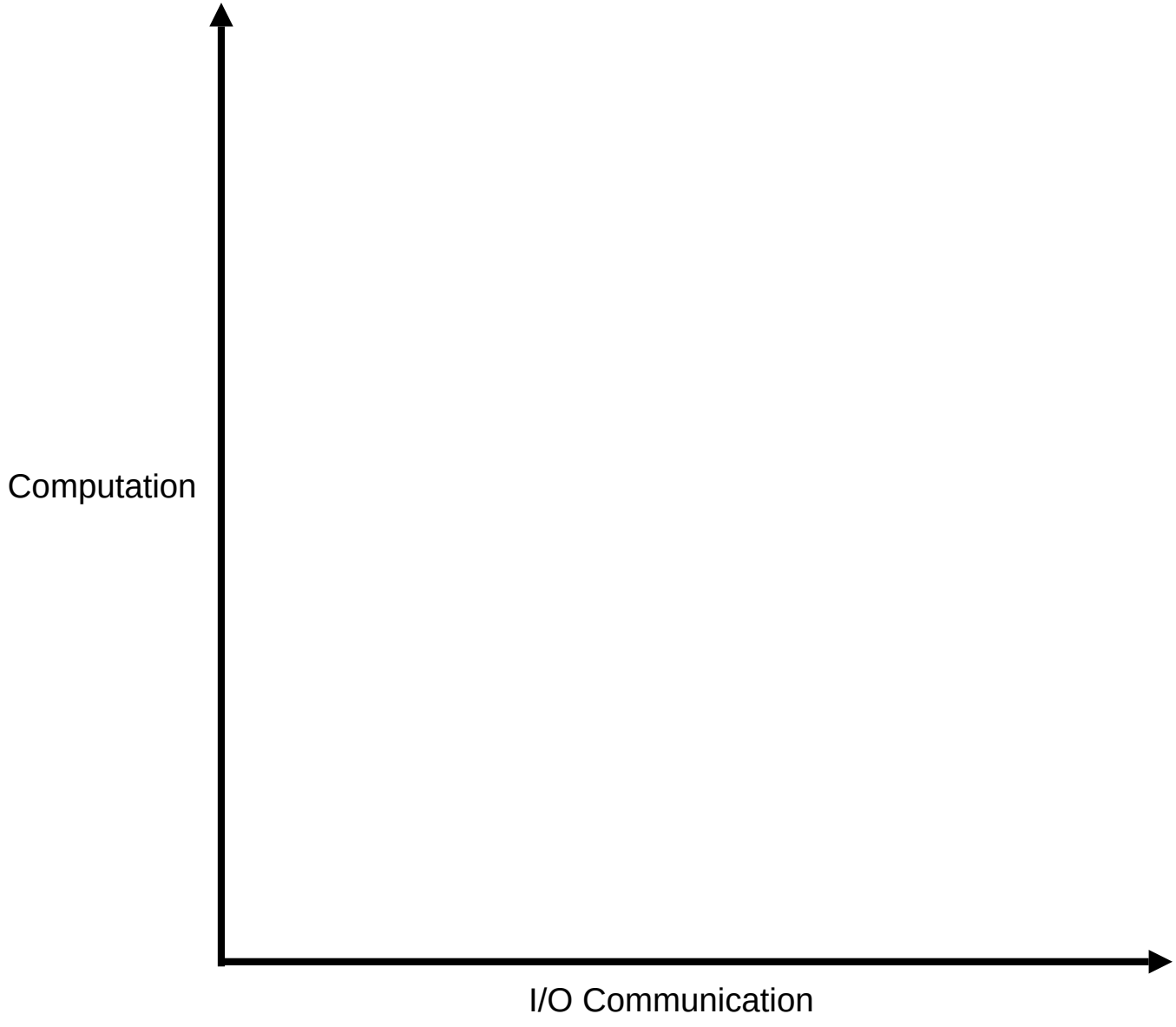
# Advantages of Software Peripherals

- Design reusability
- Easier upgrades
- Reduced manufacturing cost
- Faster design turnaround time

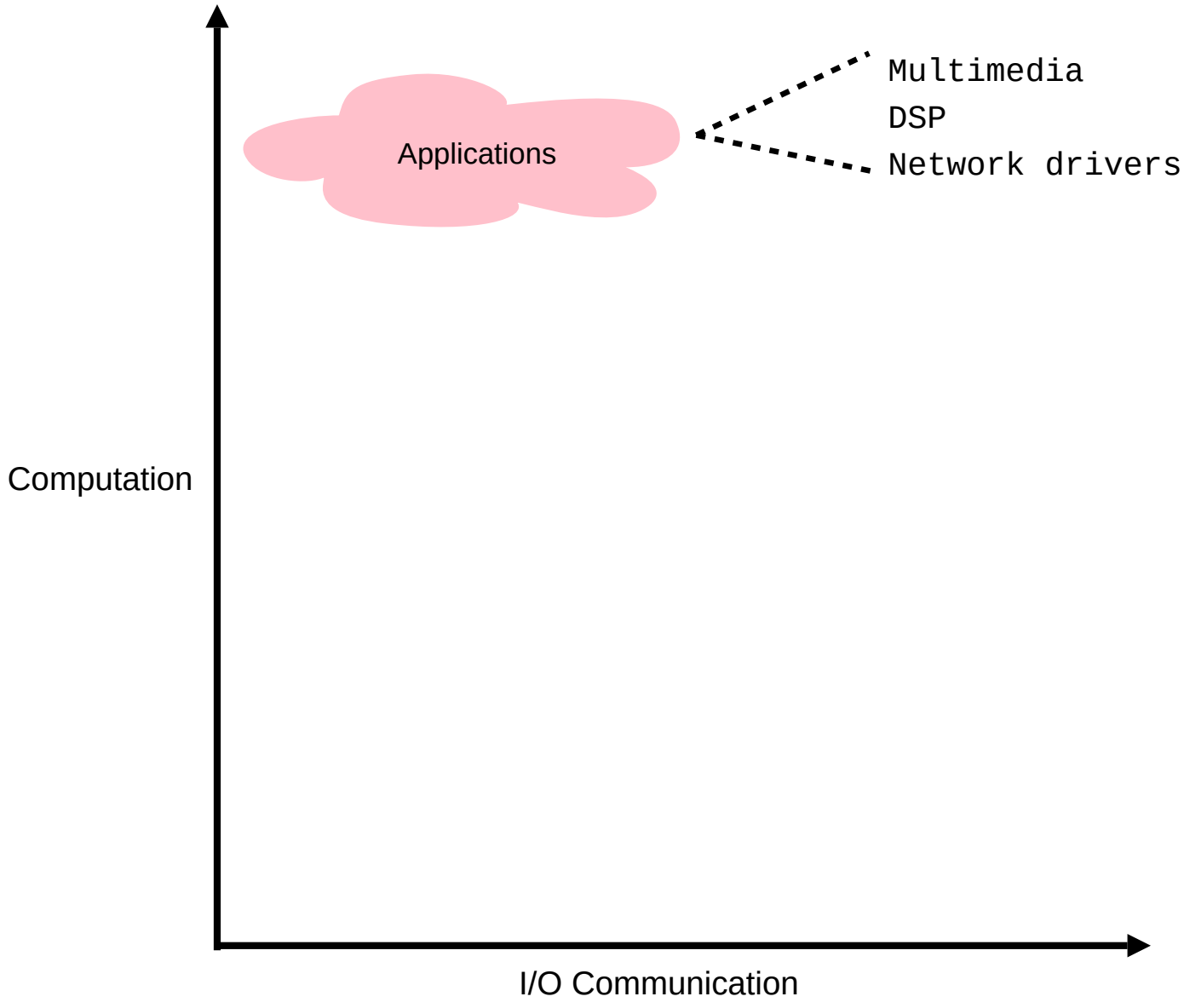
# Advantages of Software Peripherals

- Design reusability
- Easier upgrades
- Reduced manufacturing cost
- Faster design turnaround time

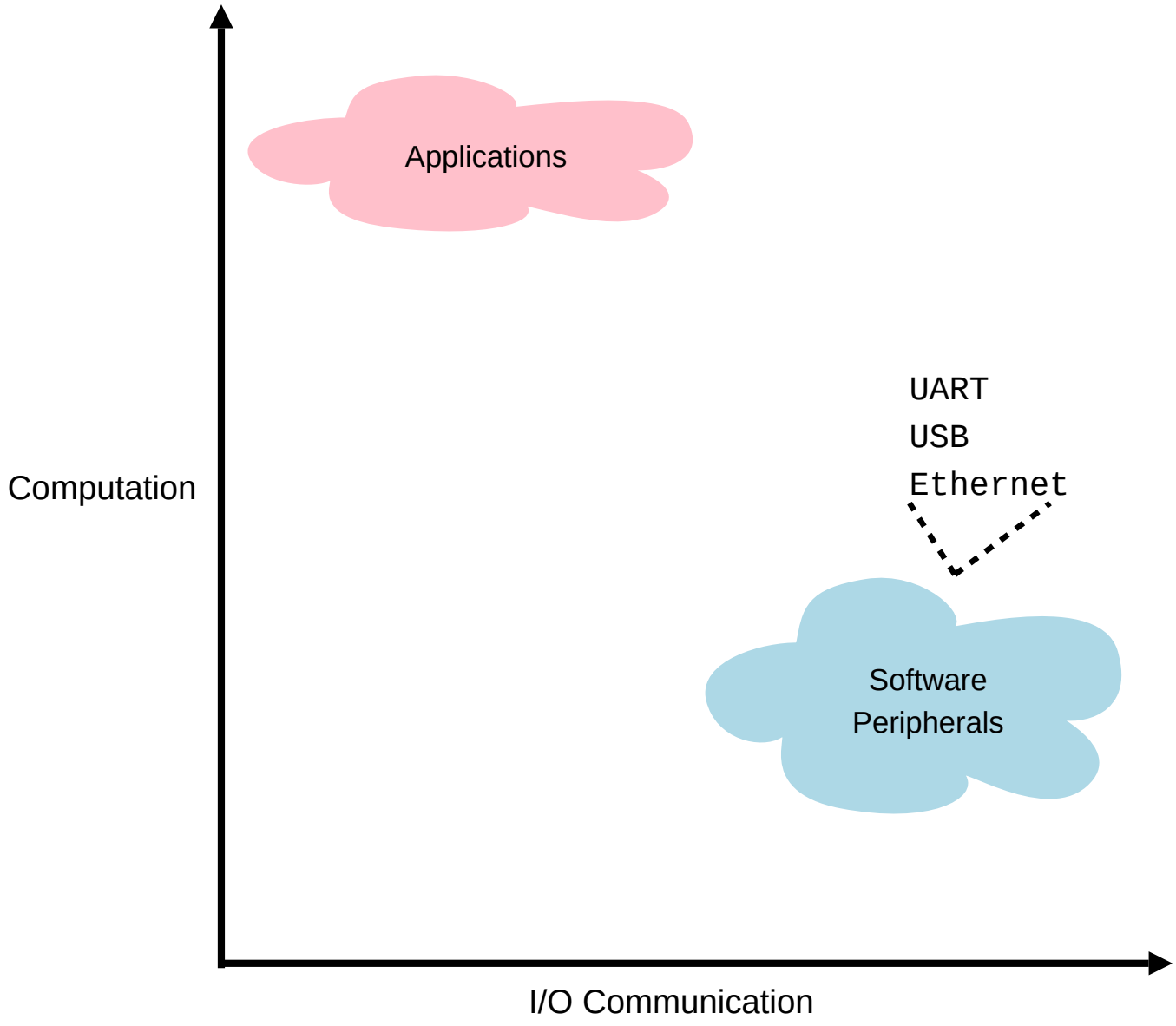
# Goals



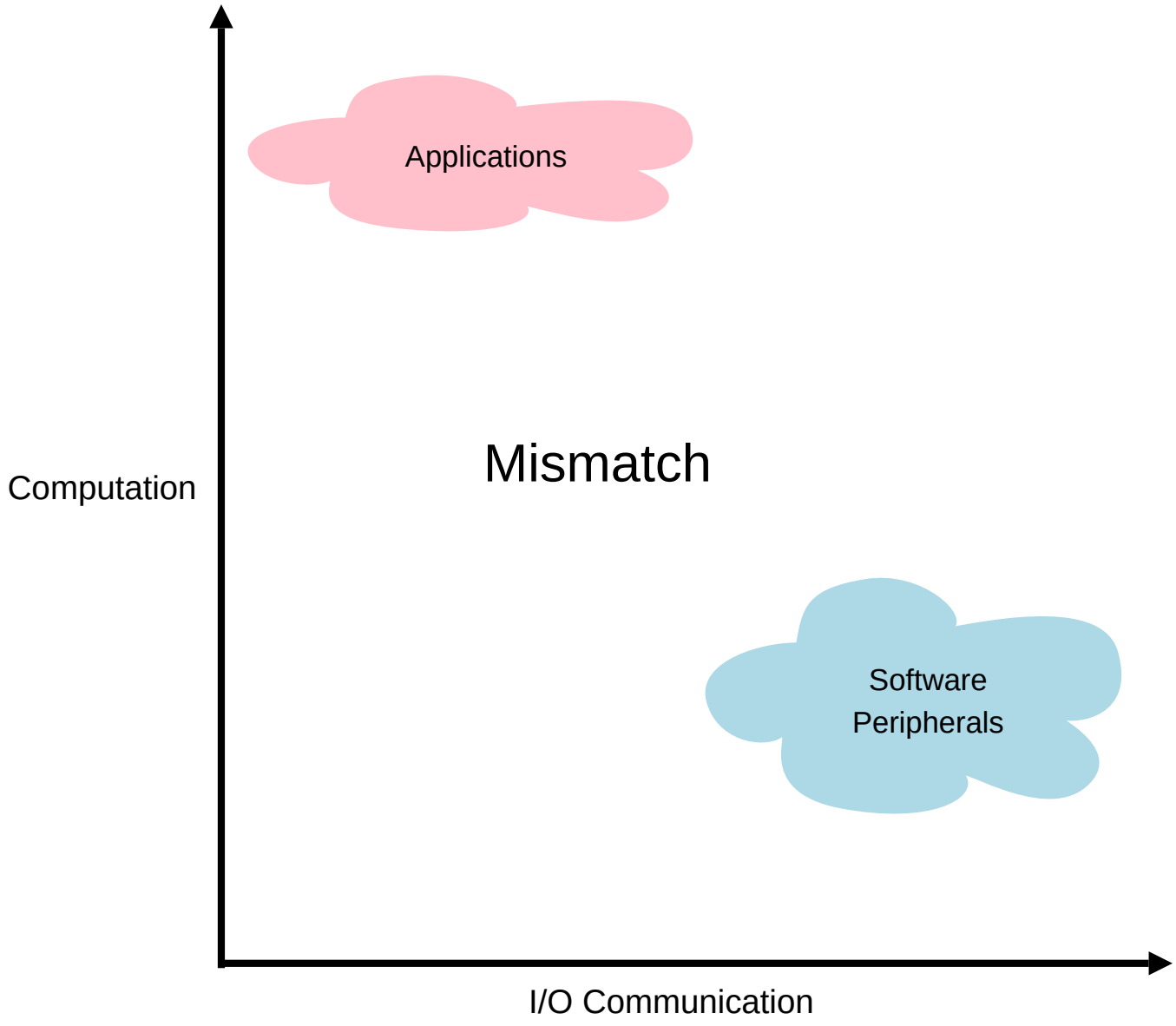
# Goals


Applications
Multimedia
DSP
Network drivers
Computation
I/O Communication

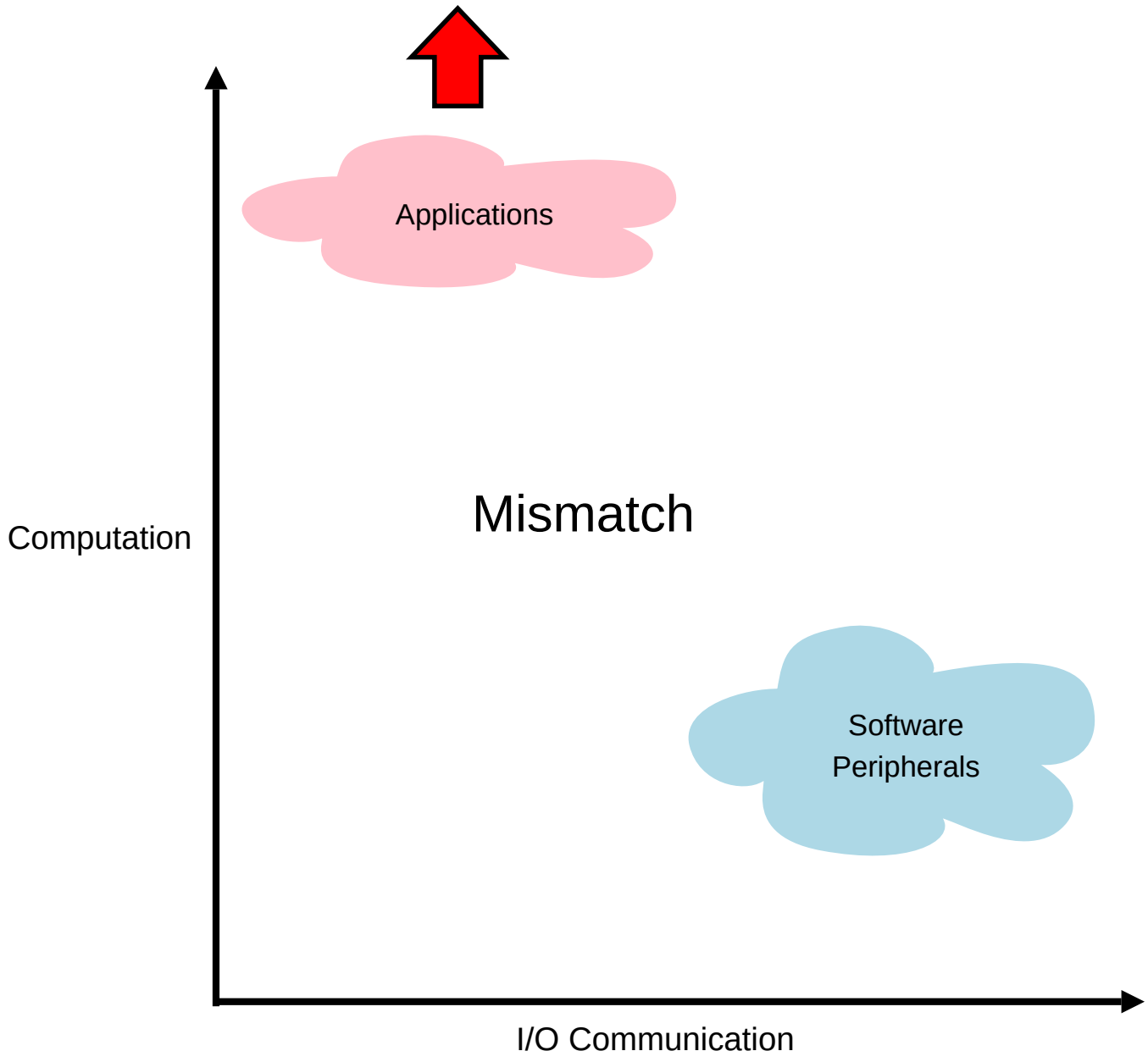
# Goals


Applications
UART
USB
Ethernet
Software
Peripherals
Computation
I/O Communication

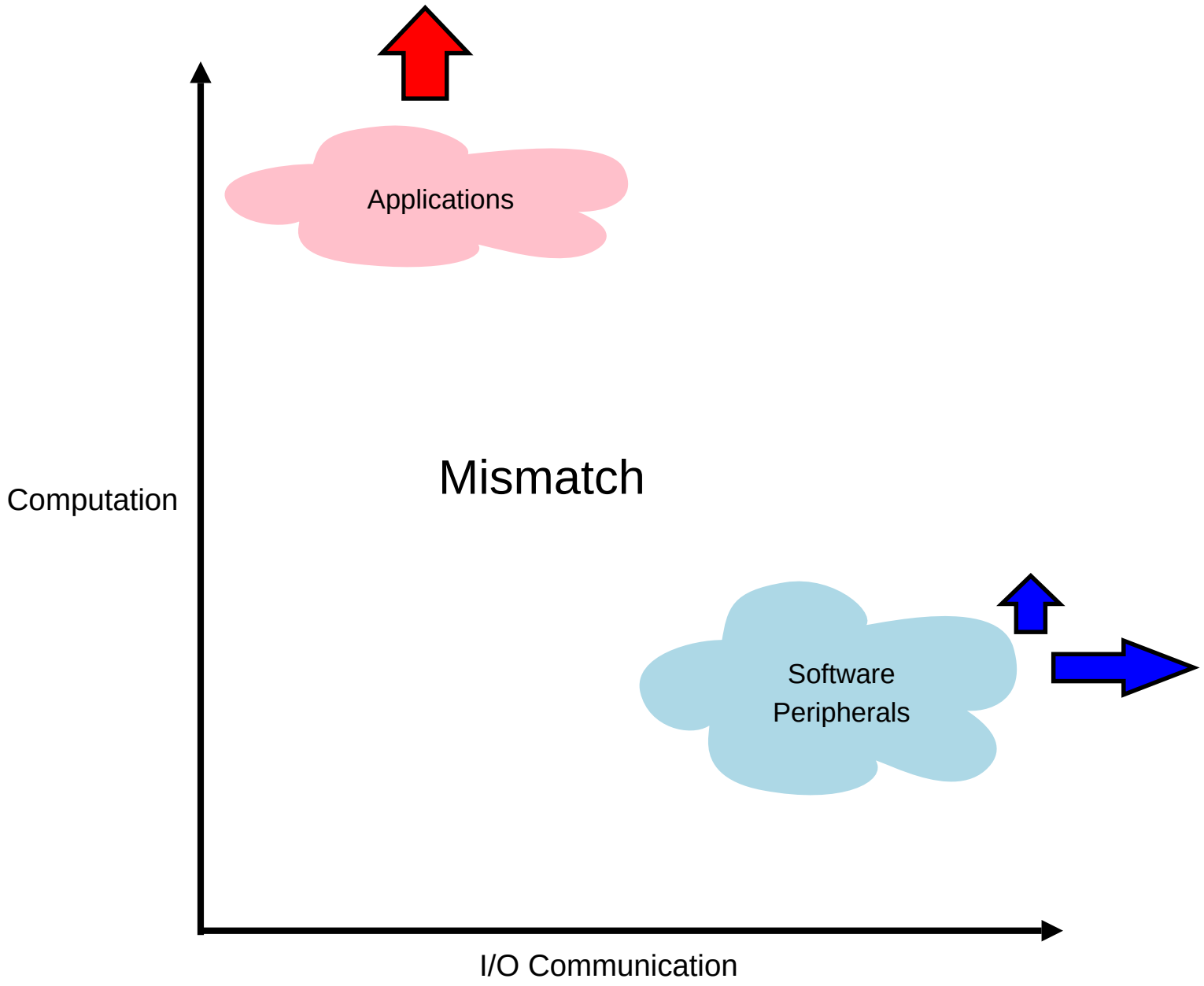
# Goals


Applications
Mismatch
Computation
Software
Peripherals
I/O Communication

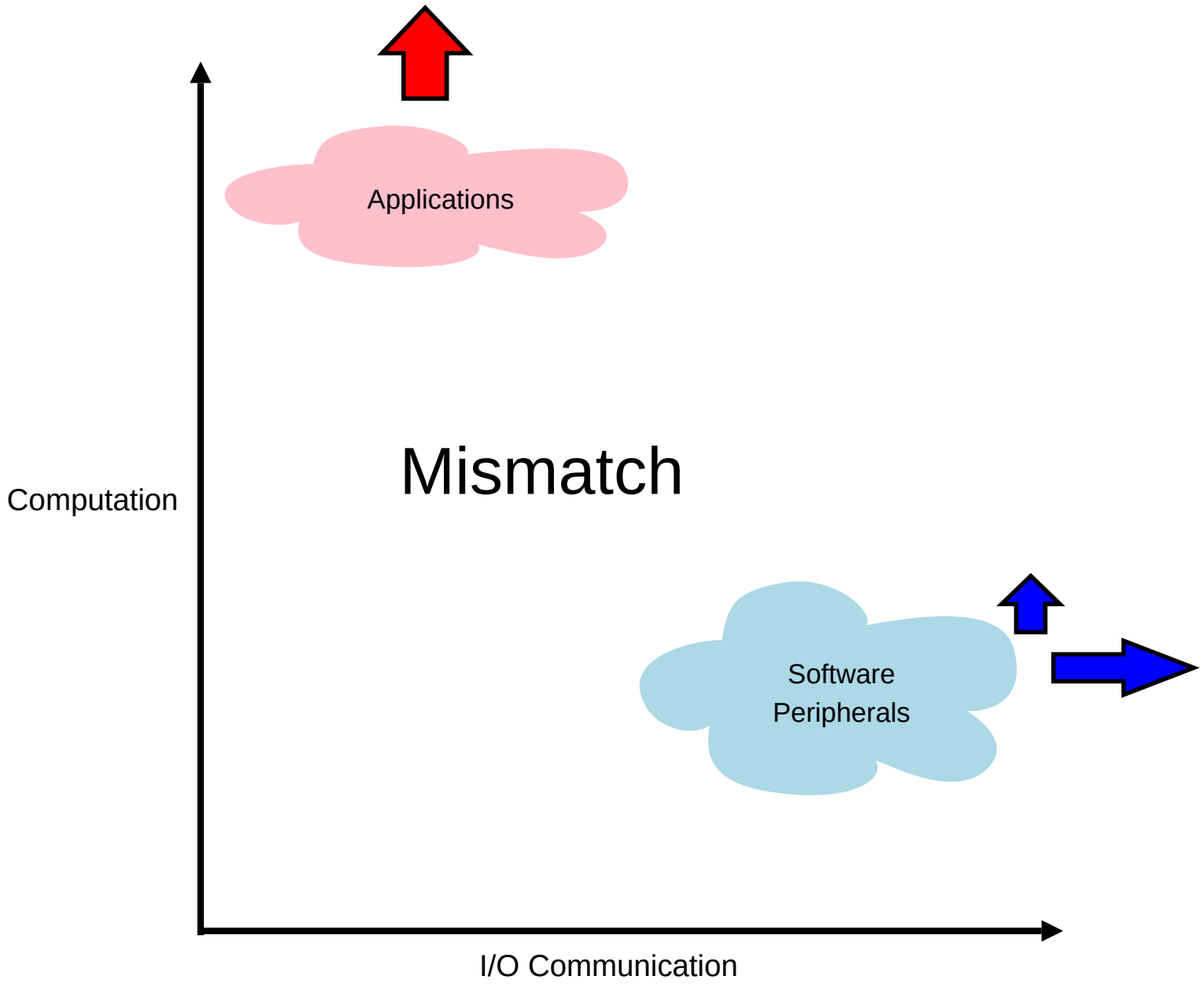
# Goals


Applications
Mismatch
Computation
Software
Peripherals
I/O Communication

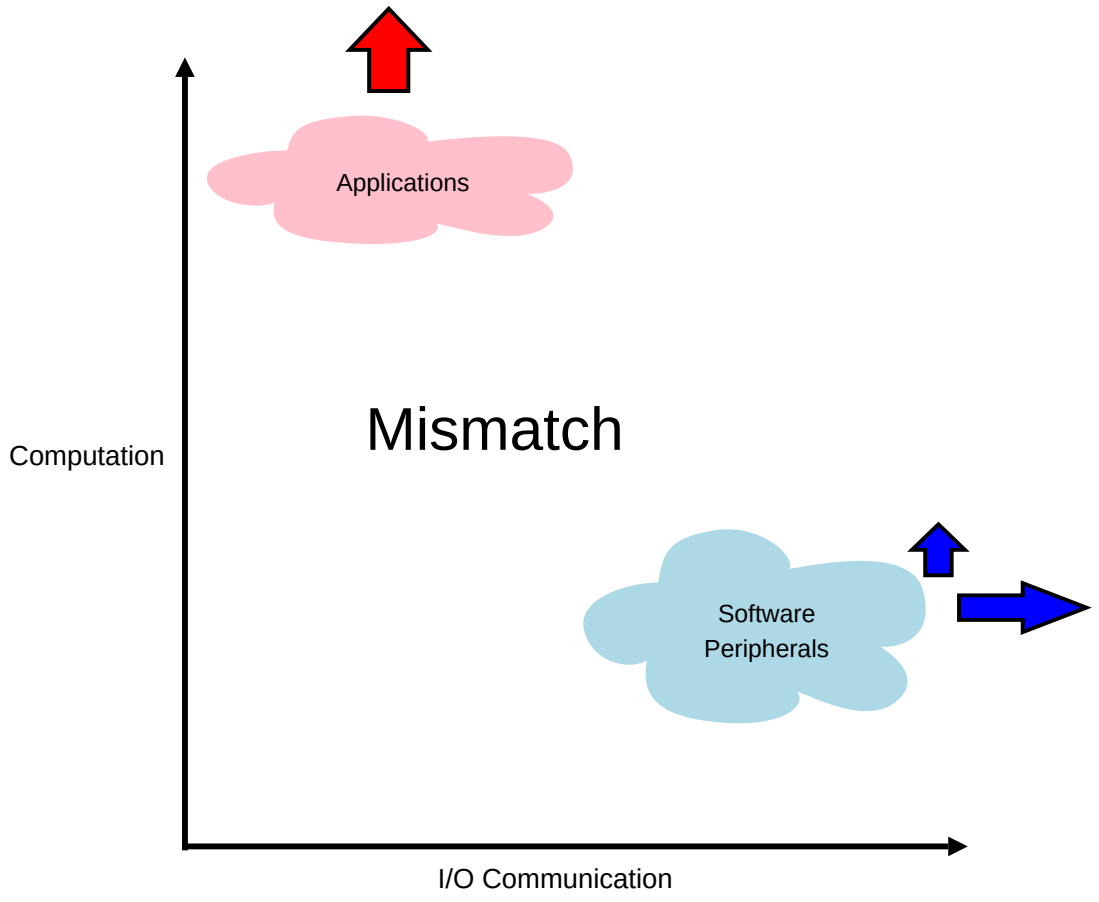
# Goals


Applications
Mismatch
Computation
Software
Peripherals
I/O Communication

# Goals


Applications
Mismatch
Computation
Software
Peripherals
I/O Communication

# Goals




*System architecture that minimizes the mismatch*

# What favors Software Peripherals ?

- Increasing speeds in embedded processors

# What favors Software Peripherals ?

- Increasing speeds in embedded processors
  - Example: ARM, PowerPC

# What favors Software Peripherals ?

- Increasing speeds in embedded processors
  - Example: ARM, PowerPC
- Processors have support for register banking, on-board A/D convertors

# What favors Software Peripherals ?

- Increasing speeds in embedded processors
  - Example: ARM, PowerPC
- Processors have support for register banking, on-board A/D convertors
  - Example: ARC processors, microcontrollers

# What favors Software Peripherals ?

- Increasing speeds in embedded processors
  - Example: ARM, PowerPC
- Processors have support for register banking, on-board A/D convertors
  - Example: ARC processors, microcontrollers
- Assumptions

# What favors Software Peripherals ?

- Increasing speeds in embedded processors
  - Example: ARM, PowerPC
- Processors have support for register banking, on-board A/D convertors
  - Example: ARC processors, microcontrollers
- Assumptions
  - Fast context switch - processors with register banking

# Related Work

- Lioupis, D., et. al,"A systematic approach to software peripherals for embedded systems", CODES-ISSS, 2001

# Related Work

- Lioupis, D., et. al,"A systematic approach to software peripherals for embedded systems", CODES-ISSS, 2001
- Ubicom `www.ubicom.com`

# Related Work

- Lioupis, D., et. al,"A systematic approach to software peripherals for embedded systems", CODES-ISSS, 2001
- Ubicom `www.ubicom.com`
  - IP 3000 family processors

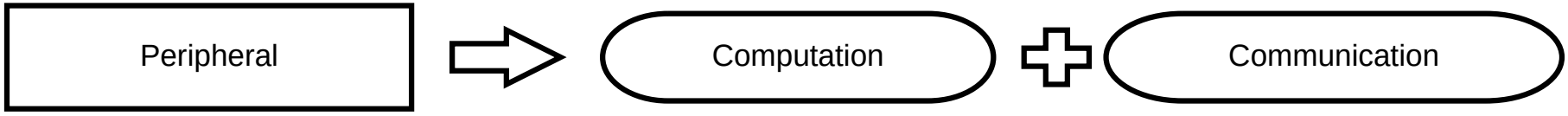
# Related Work

- Lioupis, D., et. al,"A systematic approach to software peripherals for embedded systems", CODES-ISSS, 2001
- Ubicom `www.ubicom.com`
  - IP 3000 family processors
- Triscend `www.triscend.com`

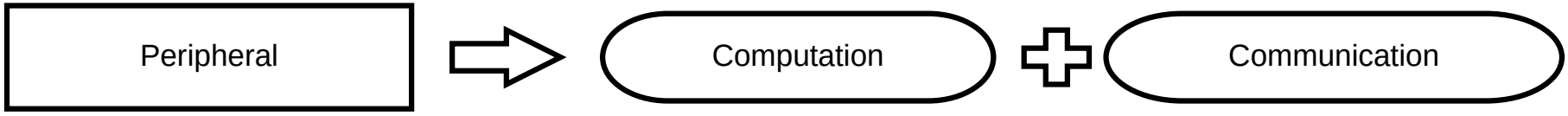
# Related Work

- Lioupis, D., et. al,"A systematic approach to software peripherals for embedded systems", CODES-ISSS, 2001
- Ubicom `www.ubicom.com`
  - IP 3000 family processors
- Triscend `www.triscend.com`
  - *Fastchip* configuration tool

# System Architecture

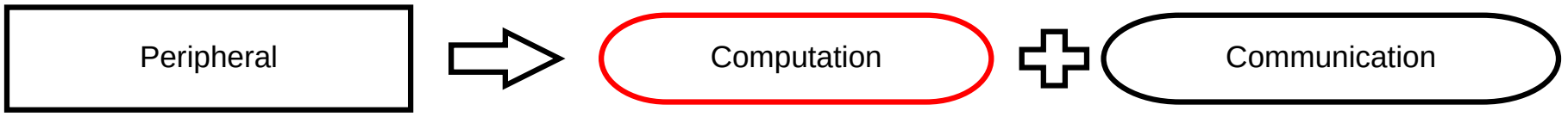


# System Architecture




- Example: RS 232 - UART
  - *8 bits, parity, 1 start-stop bit*

# System Architecture

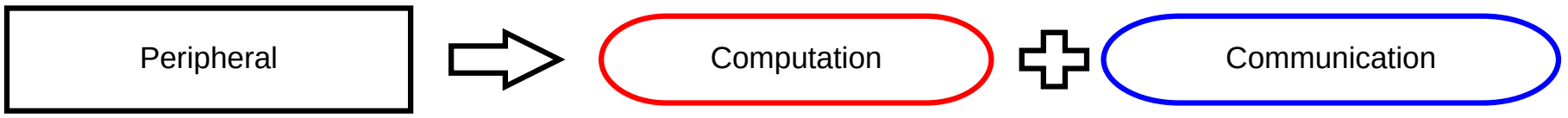



- Example: RS 232 - UART
  - *8 bits, parity, 1 start-stop bit*
- Computation
  - Serialize bits
  - Calculate Parity
  - Add start, stop bits

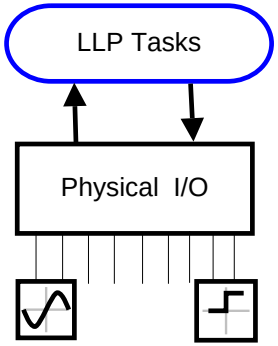
# System Architecture


- Example: RS 232 - UART
  - *8 bits, parity, 1 start-stop bit*
- Computation
  - Serialize bits
  - Calculate Parity
  - Add start, stop bits
- Communication
  - Transfer bits to Pin I/O

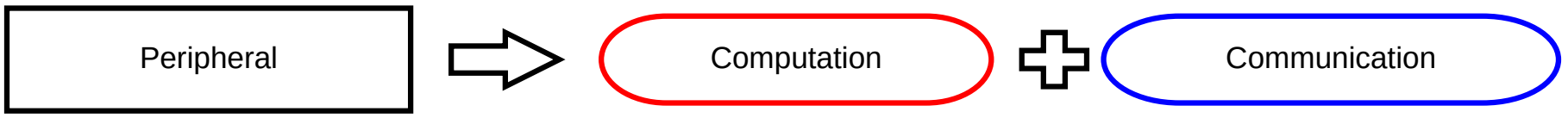
# System Architecture

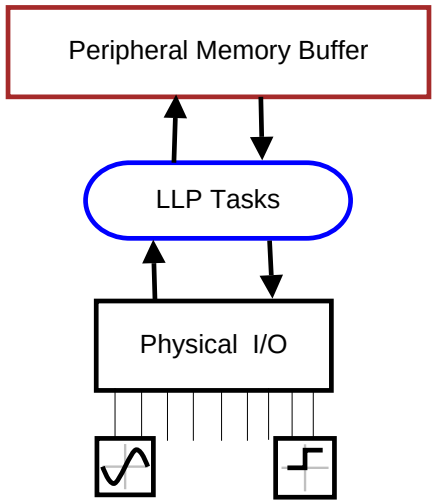



- Low Level Peripheral (LLP) Task
  - Hard real time
  - Communication intensive
  - Statically scheduled
  - As frequent as the highest operating peripheral



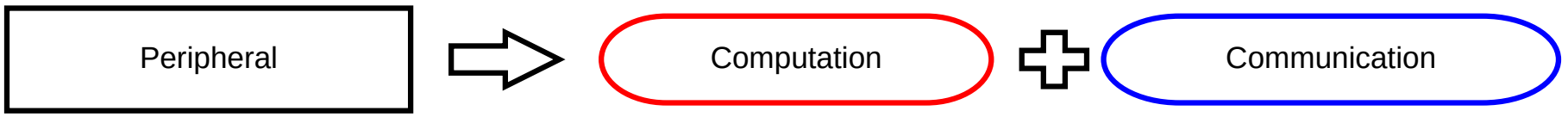
# System Architecture

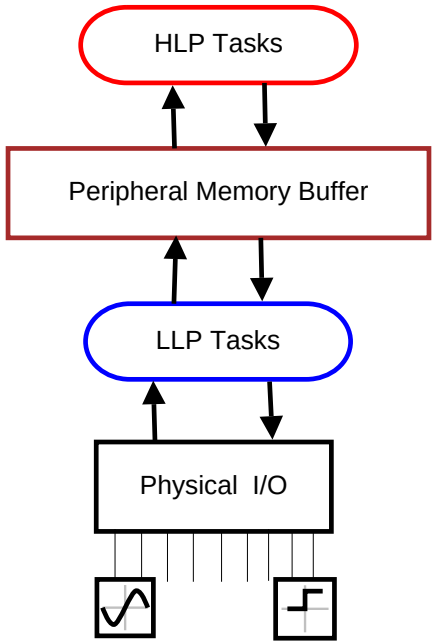



- Low Level Peripheral (LLP) Task
  - Hard real time
  - Communication intensive
  - Statically scheduled
  - As frequent as the highest operating peripheral
- Peripheral Memory Buffer
  - Shared Memory
  - Pool of buffers for each peripheral

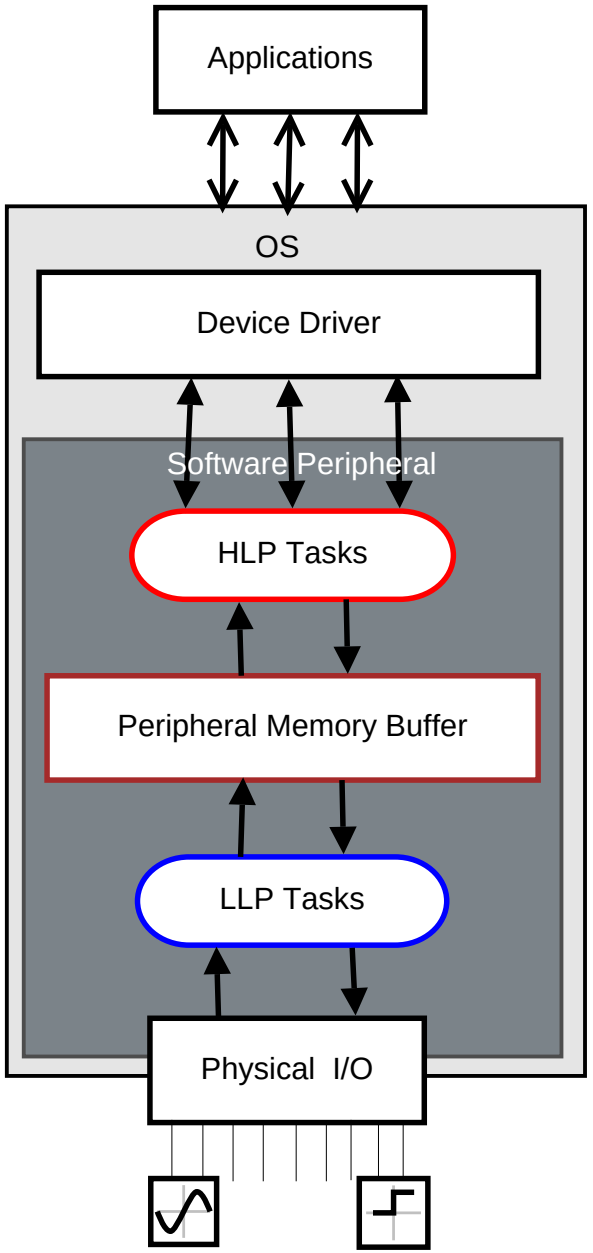
# System Architecture




- Low Level Peripheral (LLP) Task
  - Hard real time
  - Communication intensive
  - Statically scheduled
  - As frequent as the highest operating peripheral
- Peripheral Memory Buffer
  - Shared Memory
  - Pool of buffers for each peripheral
- High Level Peripheral (HLP) Task
  - Implementation of peripheral
  - Compute intensive

# System Architecture




- Low Level Peripheral (LLP) Task
  - Hard real time
  - Communication intensive
  - Statically scheduled
  - As frequent as the highest operating peripheral
- Peripheral Memory Buffer
  - Shared Memory
  - Pool of buffers for each peripheral
- High Level Peripheral (HLP) Task
  - Implementation of peripheral
  - Compute intensive

# Scheduling LLP Task

*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$

*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$

# Scheduling LLP Task

*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$

*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$
- Calculate Hyperperiod $H = LCM(T_1, T_2, ..., T_N)$

# Scheduling LLP Task

*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$

*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$
- Calculate Hyperperiod $H = LCM(T_1, T_2, ..., T_N)$
- For each permutation of task set $T_c = permute(T)$ try schedule

# Scheduling LLP Task

*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$

*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$
- Calculate Hyperperiod $H = LCM(T_1, T_2, ..., T_N)$
- For each permutation of task set $T_c = permute(T)$ try schedule
- Generate code

# Scheduling LLP Task

*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$

*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$
- Calculate Hyperperiod $H = LCM(T_1, T_2, ..., T_N)$
- For each permutation of task set $T_c = permute(T)$ try schedule
- Generate code

# Scheduling LLP Task

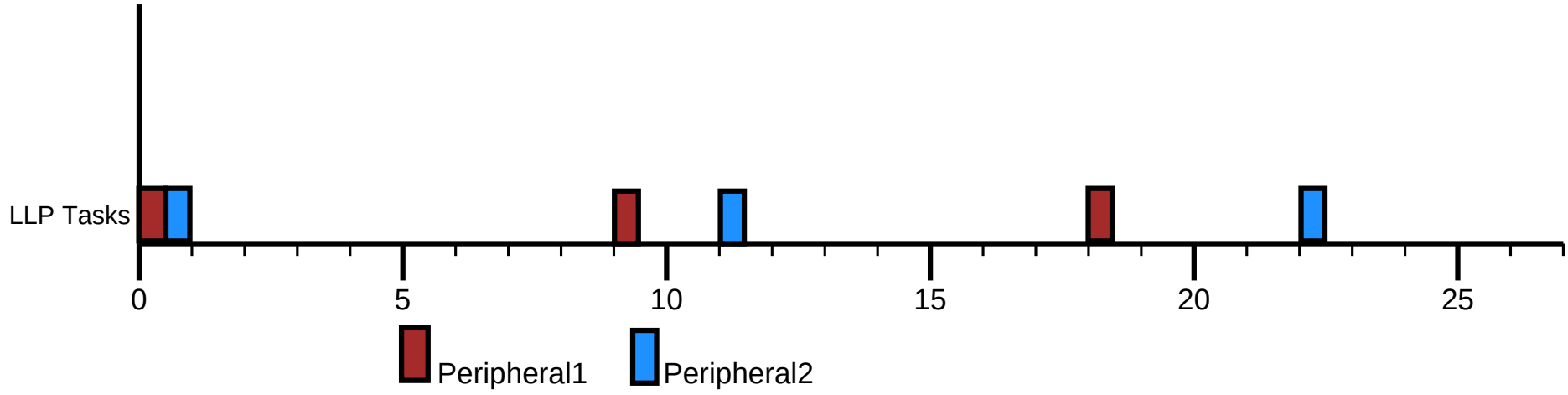
*Input* Task set $T = T_1, T_2, ..., T_N$, Context Switch Overhead $CS$
*Output* Schedule layout if successful

- Test taskset $T$ for schedulability $U > N(2^{1/N} - 1)$
- Calculate Hyperperiod $H = LCM(T_1, T_2, ..., T_N)$
- For each permutation of task set $T_c = permute(T)$ try schedule
- Generate code

Complexity $O\left((N!) \times N \times H/p_{min}\right)$
$N$ - Number of peripherals
$H$ - Hyperperiod

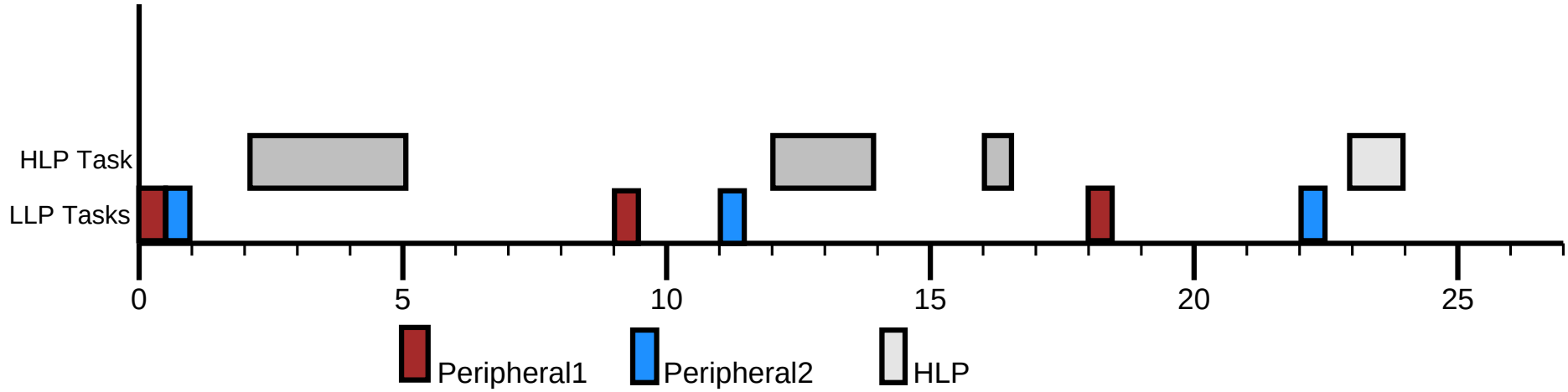
$p_{min}$ - Task with minimum periodicity

# Example Schedule




*LLP*: Hard real time

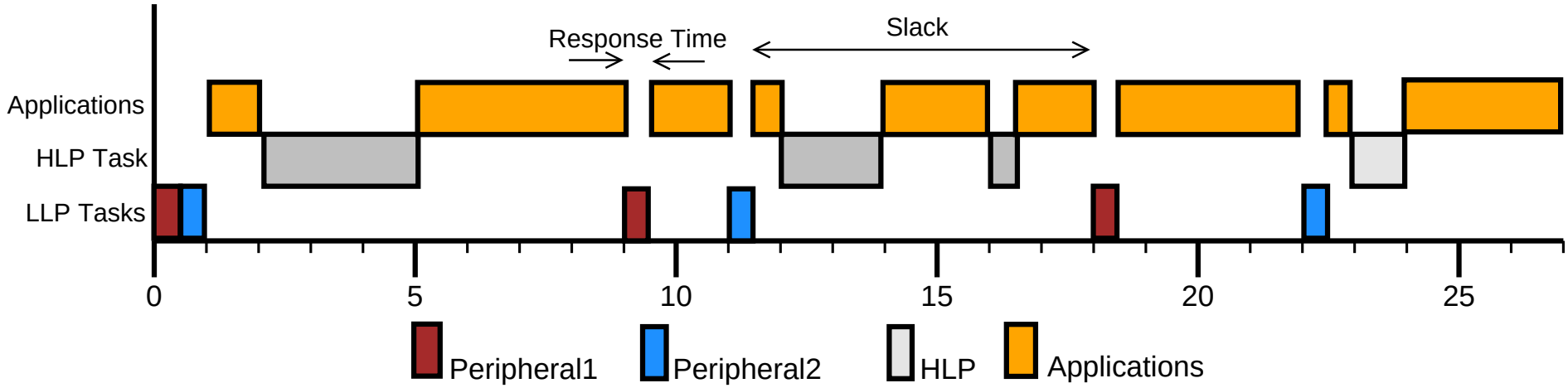
# Example Schedule




*LLP*: Hard real time

*HLP*: Soft real time

# Example Schedule




*LLP*: Hard real time

*HLP*: Soft real time

*Response Time*: Application task pre-empted due to LLP task

*Slack*: Time between two invocations of LLP task

# Simulated Peripherals

| Peripheral | HLP Cycles | LLP Cycles | Frequency | Pins |
|---|---|---|---|---|
| Serial Port | 364 | 64 | 19200 baud | 2 |
| Keypad | 16 | 29 | 1 KHz | 8 |
| Timer | 10 | 31 | 10 KHz | 1 |
| PWM | 90 | 34 | 10 KHz | 1 |
| V.34 Modem | 7660 | 32 | 33600 bps | 2 |

- *UART* 19200, 8E1 configuration

# Simulated Peripherals

| Peripheral | HLP Cycles | LLP Cycles | Frequency | Pins |
|---|---|---|---|---|
| Serial Port | 364 | 64 | 19200 baud | 2 |
| Keypad | 16 | 29 | 1 KHz | 8 |
| Timer | 10 | 31 | 10 KHz | 1 |
| PWM | 90 | 34 | 10 KHz | 1 |
| V.34 Modem | 7660 | 32 | 33600 bps | 2 |

- *UART* 19200, 8E1 configuration
- *Keypad* 4 $\times$ 4 keypad

# Simulated Peripherals

| Peripheral | HLP Cycles | LLP Cycles | Frequency | Pins |
|---|---|---|---|---|
| Serial Port | 364 | 64 | 19200 baud | 2 |
| Keypad | 16 | 29 | 1 KHz | 8 |
| Timer | 10 | 31 | 10 KHz | 1 |
| PWM | 90 | 34 | 10 KHz | 1 |
| V.34 Modem | 7660 | 32 | 33600 bps | 2 |

- *UART* 19200, 8E1 configuration
- *Keypad* 4 $\times$ 4 keypad
- *Timer* Configurable 10KHz timer

# Simulated Peripherals

| Peripheral | HLP Cycles | LLP Cycles | Frequency | Pins |
|---|---|---|---|---|
| Serial Port | 364 | 64 | 19200 baud | 2 |
| Keypad | 16 | 29 | 1 KHz | 8 |
| Timer | 10 | 31 | 10 KHz | 1 |
| PWM | 90 | 34 | 10 KHz | 1 |
| V.34 Modem | 7660 | 32 | 33600 bps | 2 |

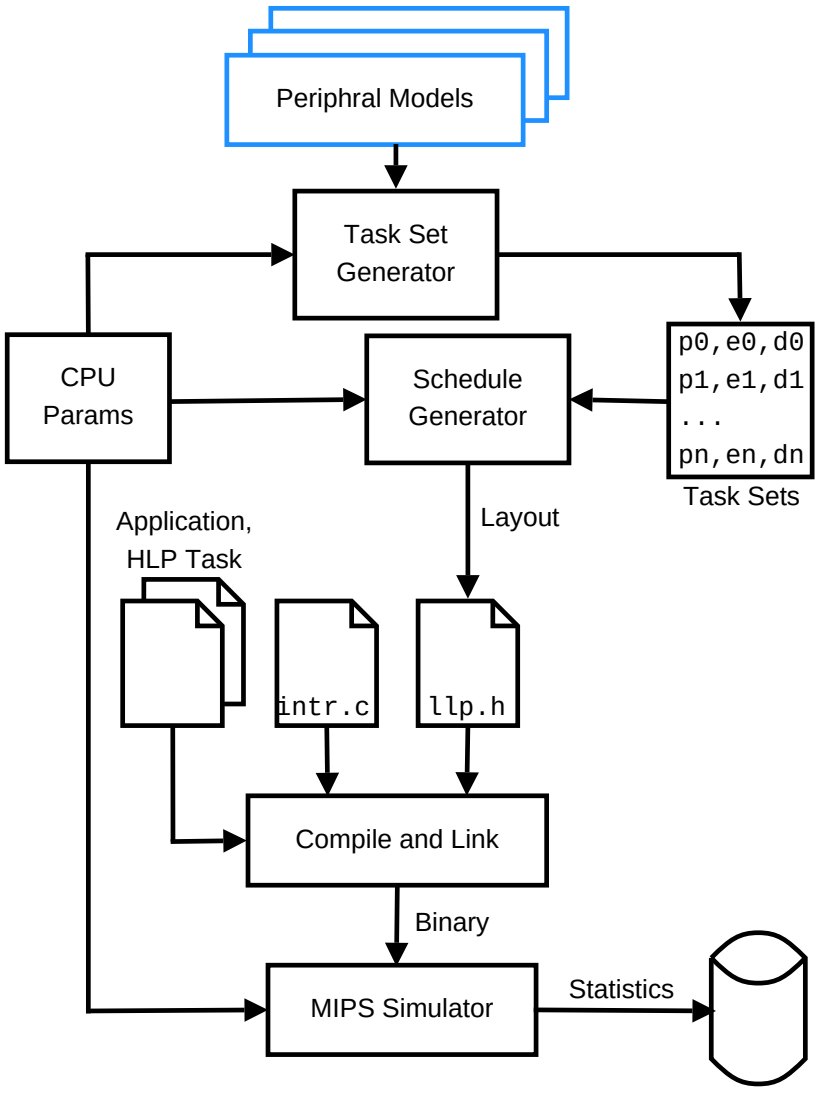
- *UART* 19200, 8E1 configuration
- *Keypad* 4 $\times$ 4 keypad
- *Timer* Configurable 10KHz timer
- *PWM* Configurable pulse width modulator

# Simulated Peripherals

| Peripheral | HLP Cycles | LLP Cycles | Frequency | Pins |
|---|---|---|---|---|
| Serial Port | 364 | 64 | 19200 baud | 2 |
| Keypad | 16 | 29 | 1 KHz | 8 |
| Timer | 10 | 31 | 10 KHz | 1 |
| PWM | 90 | 34 | 10 KHz | 1 |
| V.34 Modem | 7660 | 32 | 33600 bps | 2 |

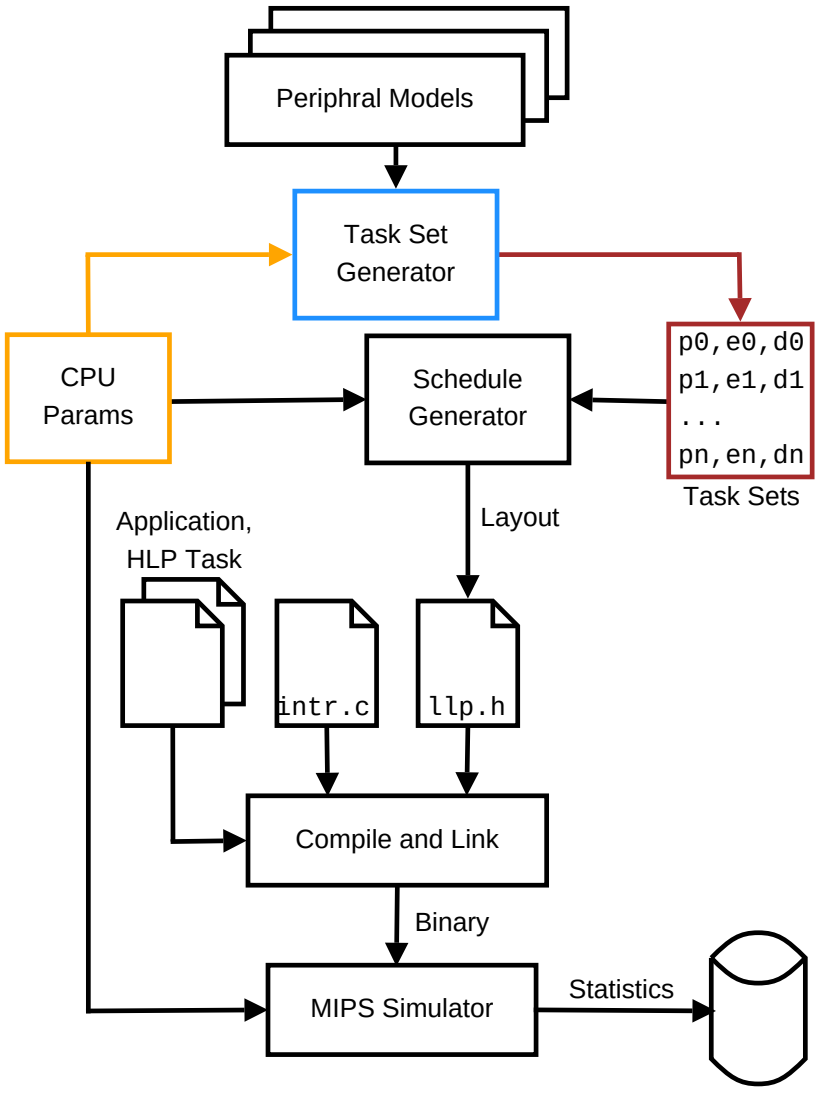
- *UART* 19200, 8E1 configuration
- *Keypad* 4 $\times$ 4 keypad
- *Timer* Configurable 10KHz timer
- *PWM* Configurable pulse width modulator
- *V.34 Modem* Implementation of V.34 protocol

# Experimental Setup



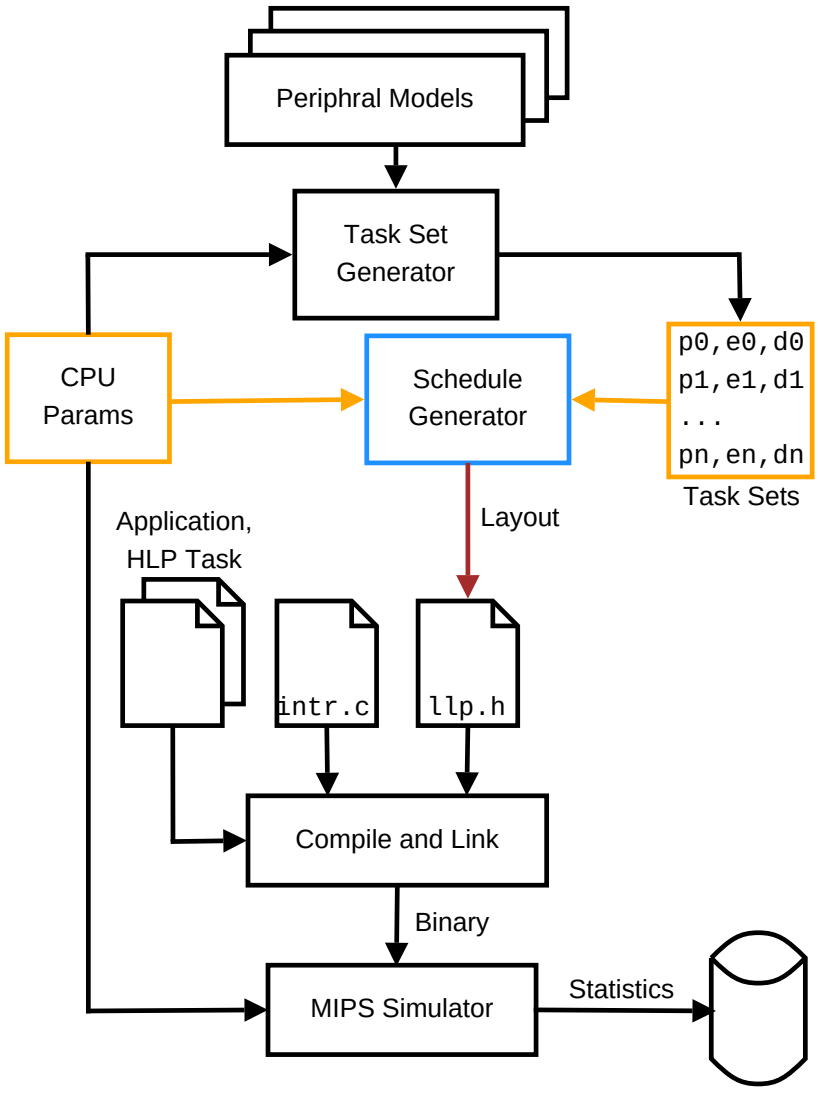

| Peripheral | HLP | LLP | Freq |
|---|---|---|---|
| Serial Port | 364 | 64 | 19200 |
| Keypad | ... | .. | ... |
| ... | | | |
| ... | | | |
| V.34 Modem | ... | .. | ... |

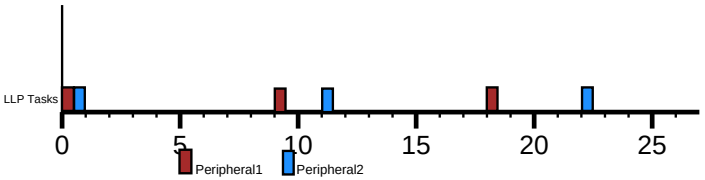
# Experimental Setup



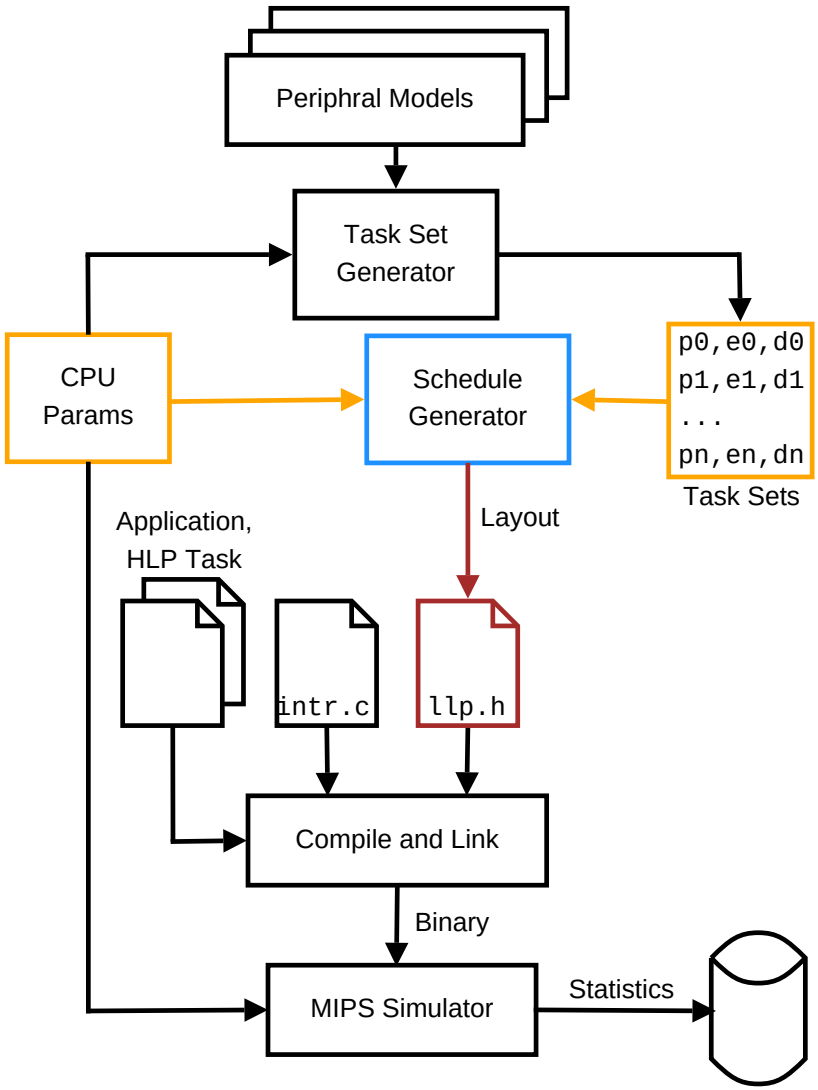

(Period, Execution Time, Deadline)
Task T1 (P1, E1, D1)
Task T2 (P2, E2, D2)
...

# Experimental Setup



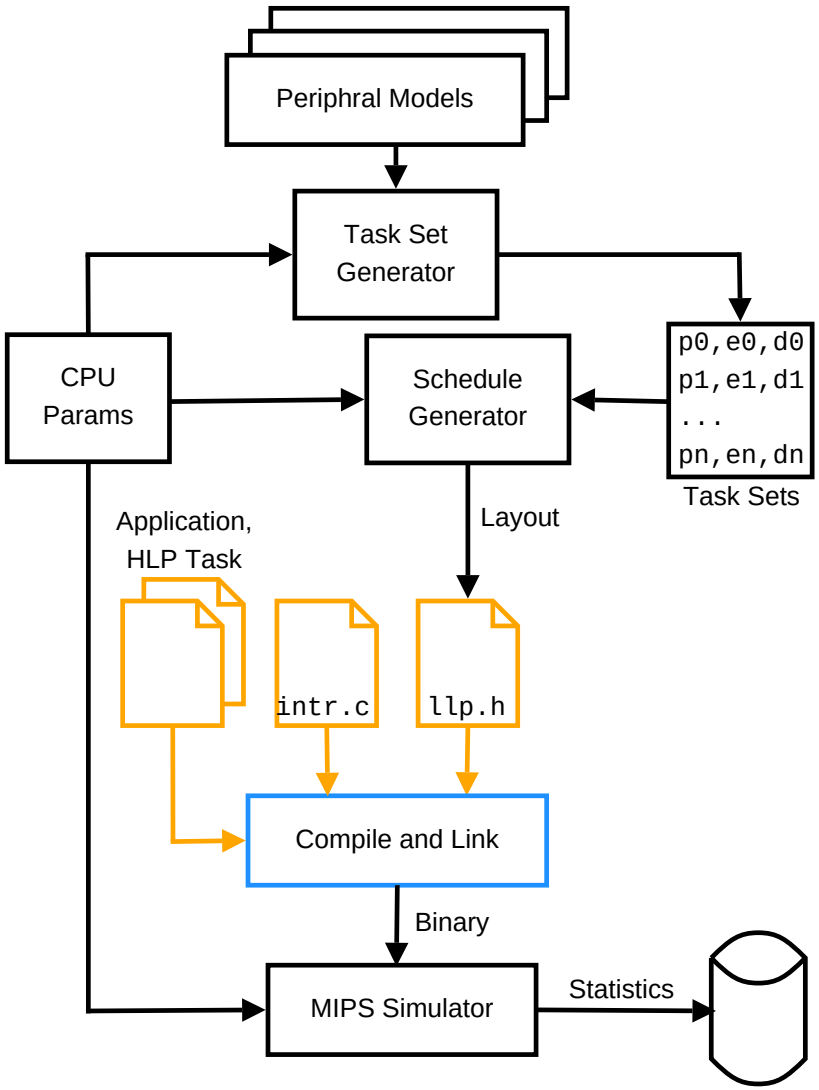



# Experimental Setup

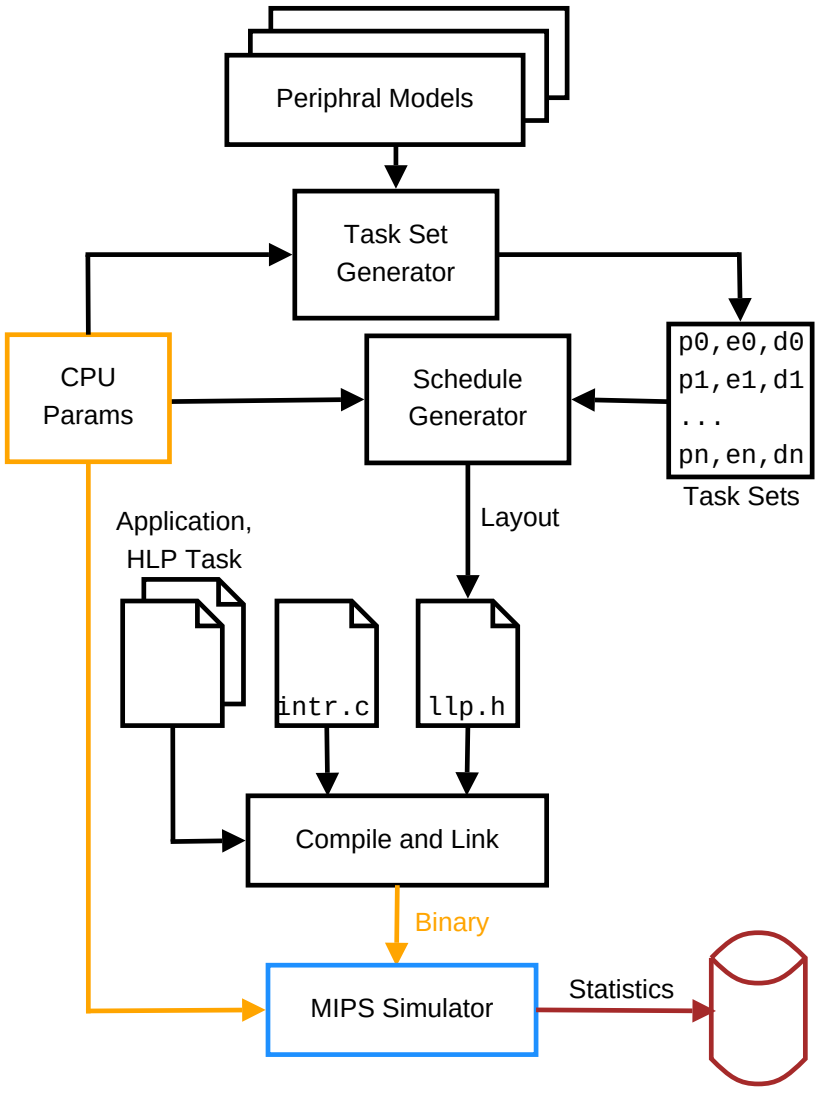



```
int T[] = {
  ...
  ...
};

int schedule[] =
{
  ...
  ...
};

#define HYPERPERIOD
#define SPI    0
#define KEYPAD 1
...
```

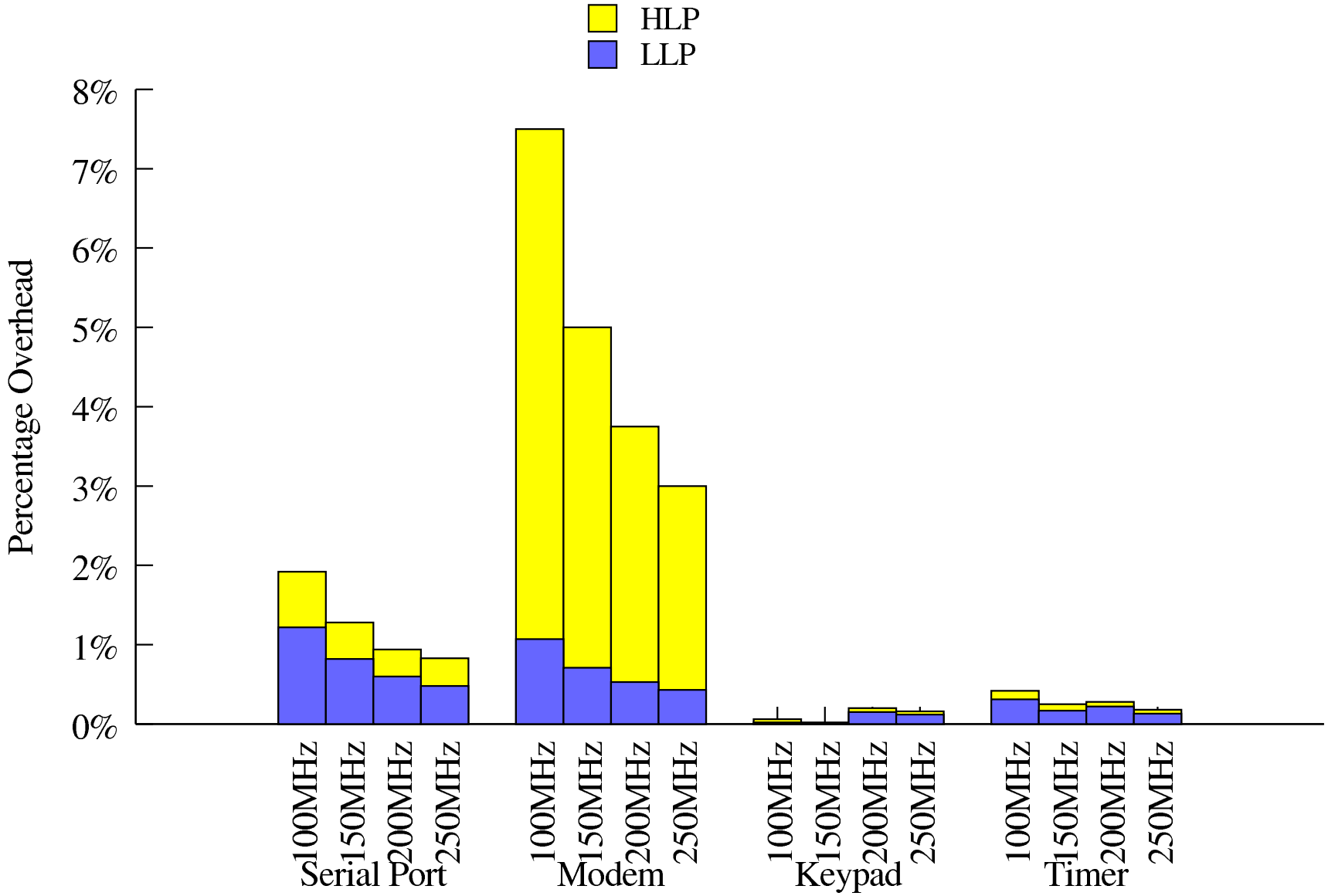
# Experimental Setup



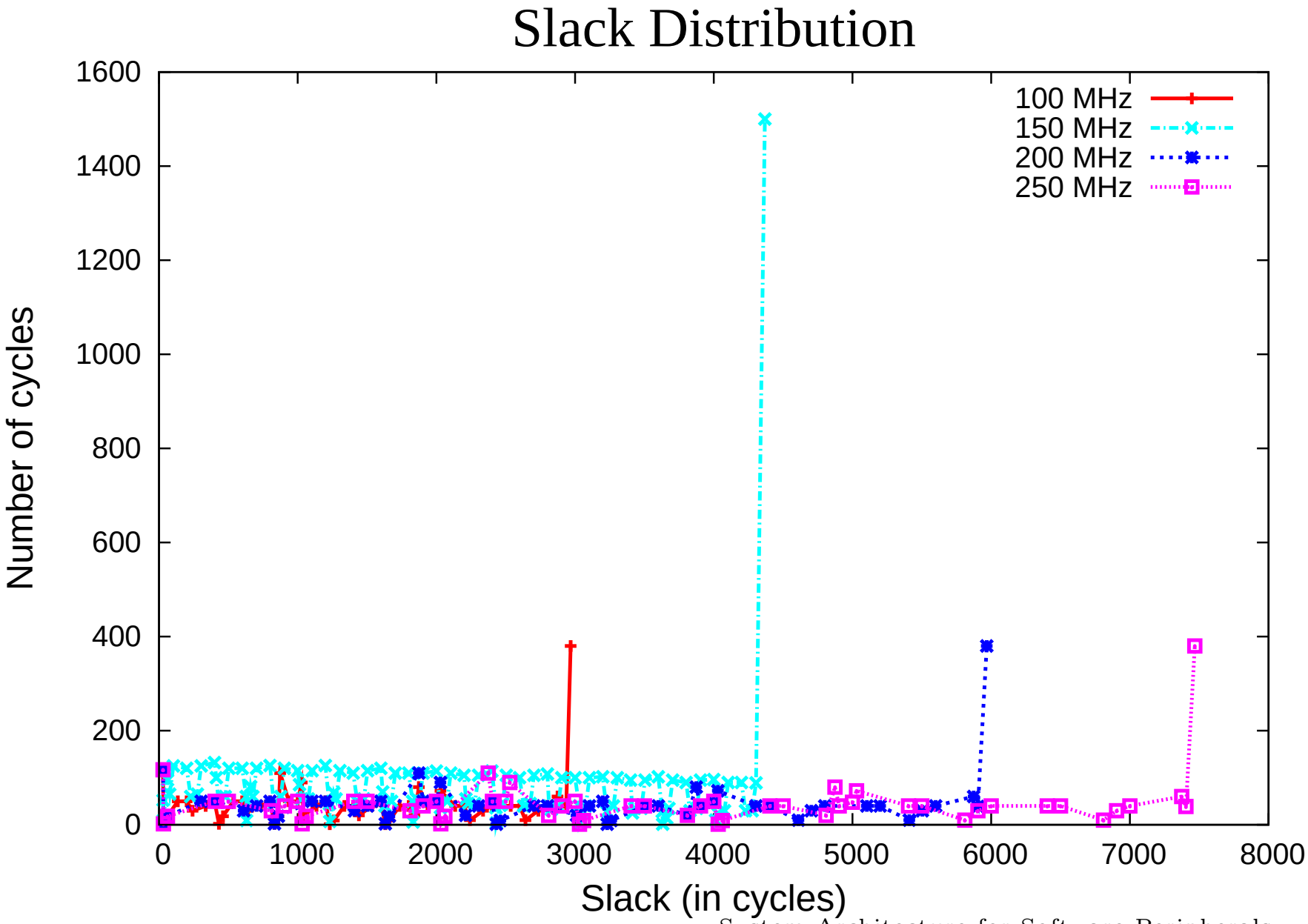
# Experimental Setup



# Results - Processor Overhead



# Results - Available Slack



# Results - Processor Utilization

| CPU (MHz) | Hyper-period | Error % | | | | | HLP (%) | LLP (%) | Intr. | Total cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial | Modem | Keypad | PWM | Timer | | | | |
| 100 | 3900000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 7.13 | 2.98 | 2869 | 3999900 |
| 150 | 21450000 | -0.10 | -1.38 | 0.0 | 0.0 | 0.0 | 4.88 | 2.08 | 10628 | 21834904 |
| 200 | 7800000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 3.60 | 1.51 | 2869 | 7899198 |
| 250 | 9750000 | -0.15 | -0.80 | 0.0 | 0.0 | 0.0 | 2.89 | 1.21 | 2869 | 9847698 |

# Results - Processor Utilization

| CPU (MHz) | Hyper-period | Error % | | | | | HLP (%) | LLP (%) | Intr. | Total cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial | Modem | Keypad | PWM | Timer | | | | |
| 100 | 3900000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 7.13 | 2.98 | 2869 | 3999900 |
| 150 | 21450000 | -0.10 | -1.38 | 0.0 | 0.0 | 0.0 | 4.88 | 2.08 | 10628 | 21834904 |
| 200 | 7800000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 3.60 | 1.51 | 2869 | 7899198 |
| 250 | 9750000 | -0.15 | -0.80 | 0.0 | 0.0 | 0.0 | 2.89 | 1.21 | 2869 | 9847698 |

# Results - Response Time

- Best case $32$ cycles

# Results - Response Time

- Best case $32$ cycles
- Worst case $190$ cycles

# Results - Response Time

- Best case $32$ cycles
- Worst case $190$ cycles
- Response time bounded by $[\min_{i=1}^{N} e_i,\ \sum_{i=1}^{N} e_i]$
  - $e_i$ - LLP task execution time of peripheral $P_i$.
  - $N$ - Number of peripherals

# Conclusion

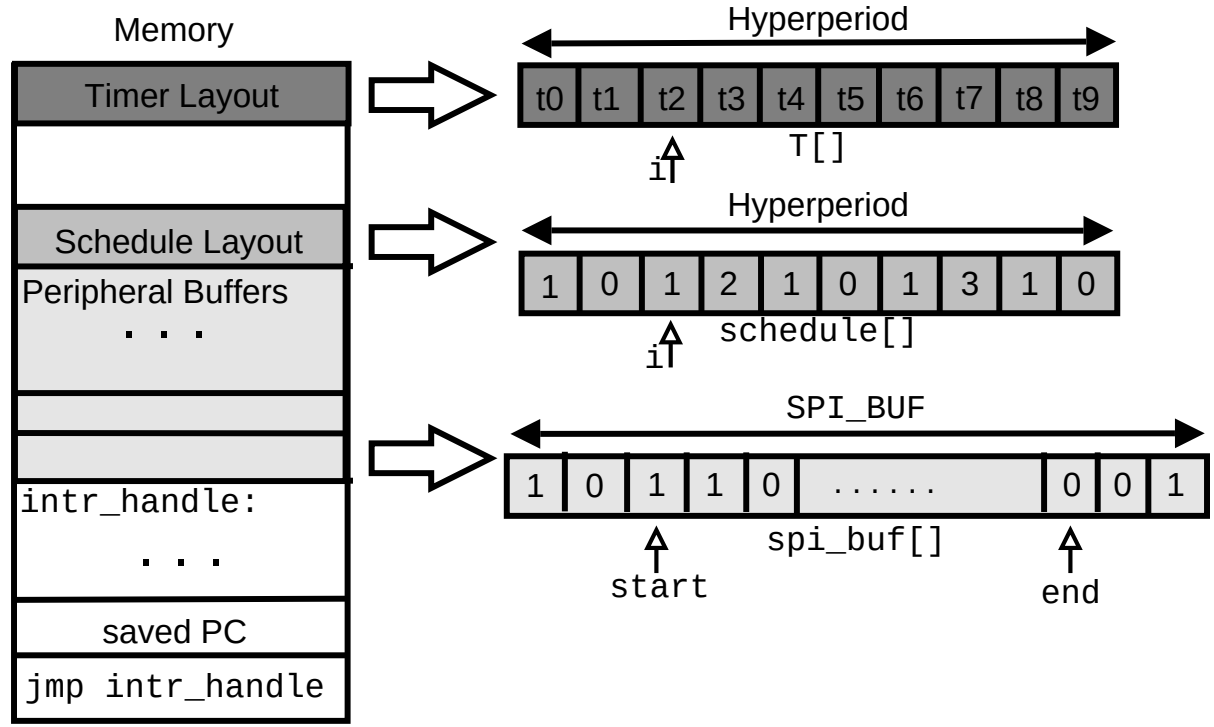
- Proposed a system architecture for software peripherals
  - Design methodology
  - Scheduling
  - Code generation
- Software Peripherals can be realized on embedded processors with right architecture support
  - Fast context switch
  - High resolution timers
  - On chip A/D support
- Minimal impact on user level applications

# Thank You

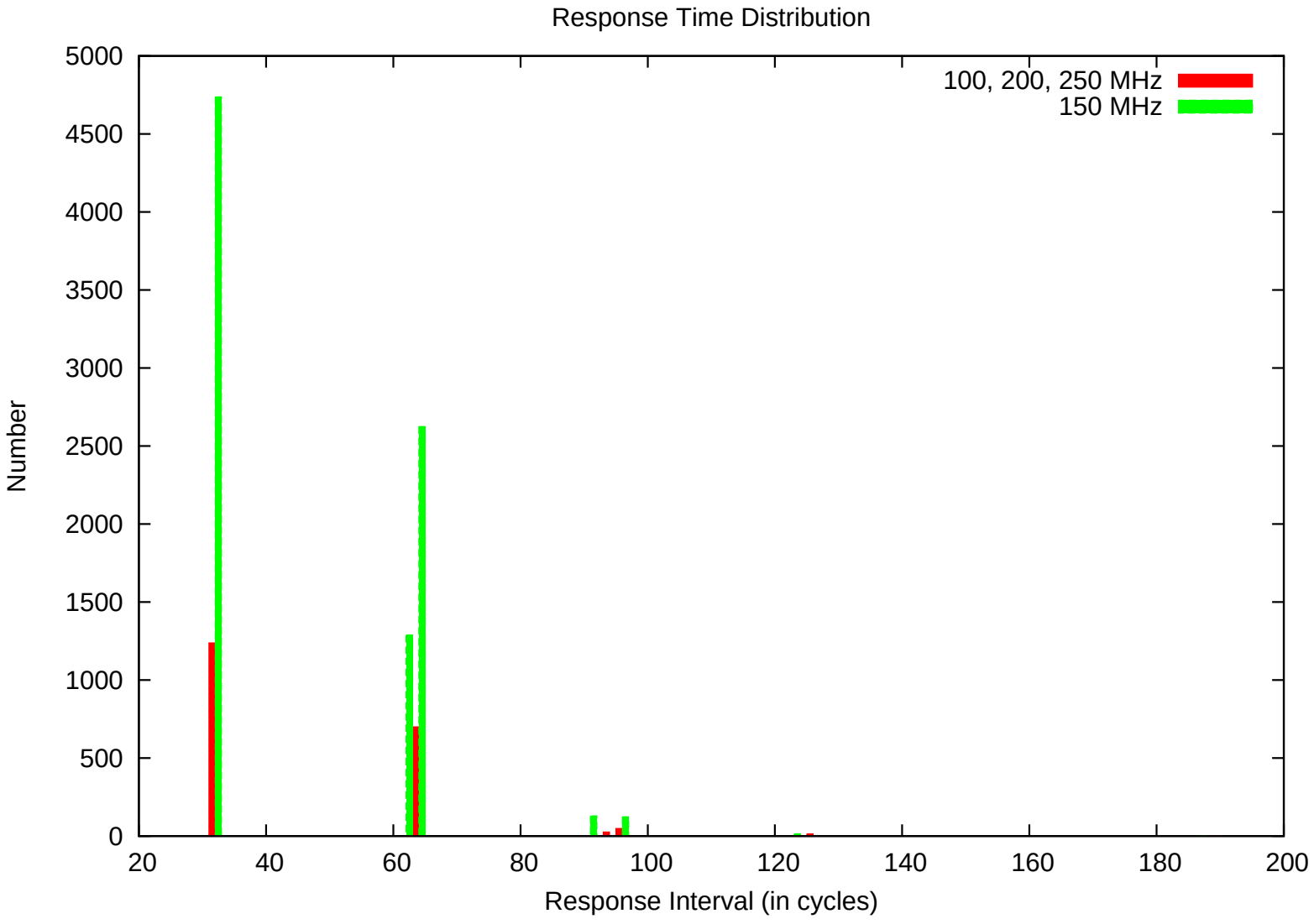
# Backup slides

# Memory Layout

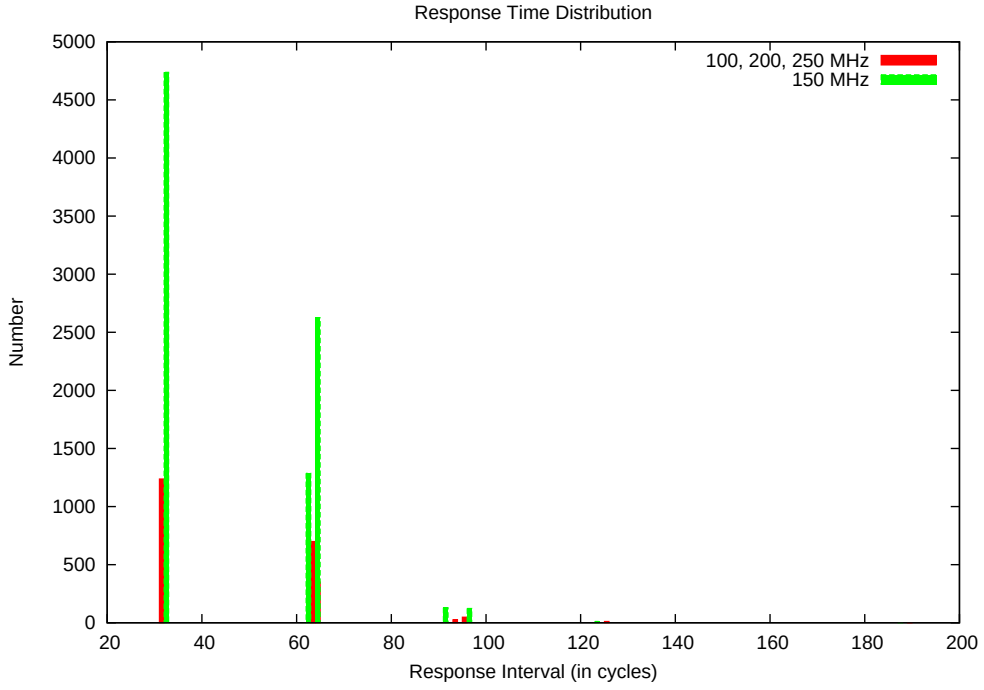
```
intr_handle() {
  switch(schedule[i]) {
    case SPI:    /* 0 */
      ...
      outp(spi_buf[start]);
      start = (start+1)%SPI_BUF;
      break;
    case KEYPAD: /* 1 */
      ...
      ...          ...
      ...
    case PWM:    /* 3 */
      ...
  }
  i = (i + 1) % HYPERPERIOD
  set_timer(T[i] - T[i-1]);
 }
```



# Results - Response Time



# Results - Response Time




- Best case $32$ cycles
- Worst case $190$ cycles
- Response time bounded by $[\min_{i=1}^{N} e_i,\ \sum_{i=1}^{N} e_i]$
  - $e_i$ - LLP task execution time of peripheral $P_i$.
  - $N$ - Number of peripherals

# Results - Processor Overhead

| **Peripheral** | **100MHz** | | | **150MHz** | | | **200MHz** | | | **250MHz** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LLP | HLP | Total | LLP | HLP | Total | LLP | HLP | Total | LLP | HLP | Total |
| Serial Port | 1.22 | 0.70 | 1.92 | 0.82 | 0.46 | 1.28 | 0.60 | 0.34 | 0.94 | 0.48 | 0.35 | 0.83 |
| Modem | 1.07 | 6.43 | 7.50 | 0.71 | 4.29 | 5.00 | 0.53 | 3.22 | 3.75 | 0.43 | 2.57 | 3.00 |
| Keypad | 0.02 | 0.02 | 0.04 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 |
| Timer | 0.31 | 0.10 | 0.41 | 0.20 | 0.07 | 0.27 | 0.15 | 0.05 | 0.20 | 0.12 | 0.04 | 0.16 |
| PWM | 0.34 | 0.11 | 0.45 | 0.17 | 0.08 | 0.25 | 0.22 | 0.06 | 0.28 | 0.13 | 0.05 | 0.18 |

# Results - Processor Overhead

| Peripheral | 100MHz | | | 150MHz | | | 200MHz | | | 250MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LLP | HLP | Total | LLP | HLP | Total | LLP | HLP | Total | LLP | HLP | Total |
| Serial Port | 1.22 | 0.70 | 1.92 | 0.82 | 0.46 | 1.28 | 0.60 | 0.34 | 0.94 | 0.48 | 0.35 | 0.83 |
| Modem | 1.07 | 6.43 | 7.50 | 0.71 | 4.29 | 5.00 | 0.53 | 3.22 | 3.75 | 0.43 | 2.57 | 3.00 |
| Keypad | 0.02 | 0.02 | 0.04 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 | 0.01 | 0.01 | 0.02 |
| Timer | 0.31 | 0.10 | 0.41 | 0.20 | 0.07 | 0.27 | 0.15 | 0.05 | 0.20 | 0.12 | 0.04 | 0.16 |
| PWM | 0.34 | 0.11 | 0.45 | 0.17 | 0.08 | 0.25 | 0.22 | 0.06 | 0.28 | 0.13 | 0.05 | 0.18 |

# Results - Processor Utilization

| CPU (MHz) | Hyper-period | Error % | | | | | HLP (%) | LLP (%) | Intr. | Total cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial | Modem | Keypad | PWM | Timer | | | | |
| 100 | 3900000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 7.13 | 2.98 | 2869 | 3999900 |
| 150 | 21450000 | -0.10 | -1.38 | 0.0 | 0.0 | 0.0 | 4.88 | 2.08 | 10628 | 21834904 |
| 200 | 7800000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 3.60 | 1.51 | 2869 | 7899198 |
| 250 | 9750000 | -0.15 | -0.80 | 0.0 | 0.0 | 0.0 | 2.89 | 1.21 | 2869 | 9847698 |

# Results - Processor Utilization

| CPU (MHz) | Hyper-period | Error % | | | | | HLP (%) | LLP (%) | Intr. | Total cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Serial | Modem | Keypad | PWM | Timer | | | | |
| 100 | 3900000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 7.13 | 2.98 | 2869 | 3999900 |
| 150 | 21450000 | -0.10 | -1.38 | 0.0 | 0.0 | 0.0 | 4.88 | 2.08 | 10628 | 21834904 |
| 200 | 7800000 | -0.15 | 0.80 | 0.0 | 0.0 | 0.0 | 3.60 | 1.51 | 2869 | 7899198 |
| 250 | 9750000 | -0.15 | -0.80 | 0.0 | 0.0 | 0.0 | 2.89 | 1.21 | 2869 | 9847698 |