THE POSSIBILITIES ARE INFINITE FUJITSU

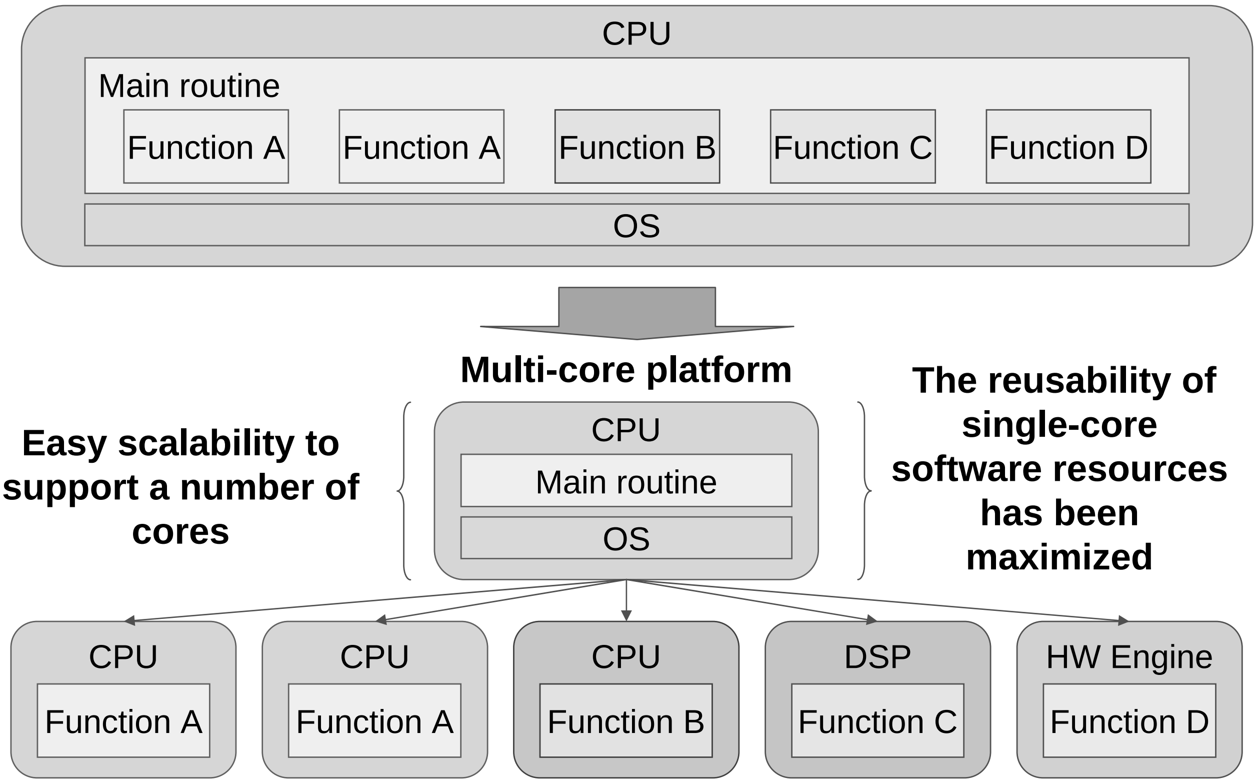
ARPC based heterogeneous multi-core platform suitable for an embedded system

Atsuhiro Suga
FUJITSU LABORATORIES LIMITED
Processor Solution Development Laboratories,
System LSI Development Laboratories

Embedded multi-core programming model

FUJITSU

Single-core platform



CPU

Main routine

Function A | Function A | Function B | Function C | Function D

OS

Multi-core platform

CPU

Main routine

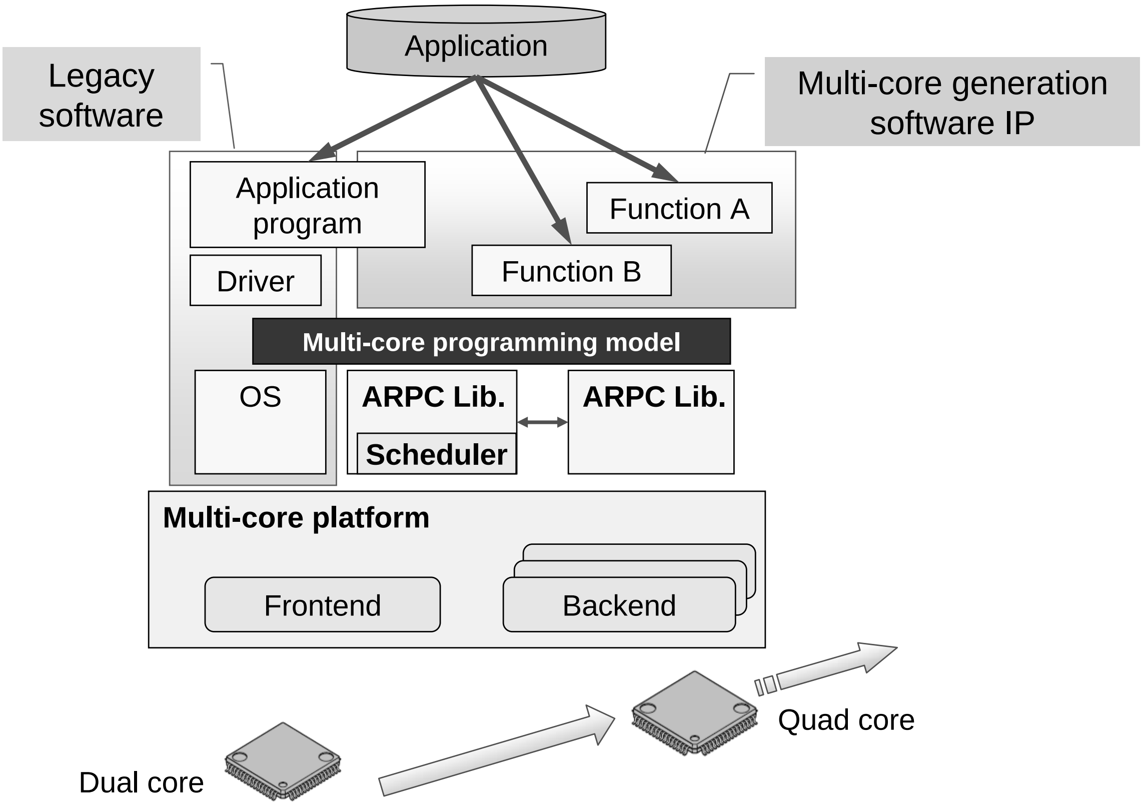
OS

CPU: Function A | CPU: Function A | CPU: Function B | DSP: Function C | HW Engine: Function D

Easy scalability to support a number of cores

The reusability of single-core software resources has been maximized

Fujitsu embedded multi-core platform



Application

Legacy software

Multi-core generation software IP

Application program

Driver

Function A

Function B

Multi-core programming model

OS

ARPC Lib.

Scheduler

ARPC Lib.

Multi-core platform

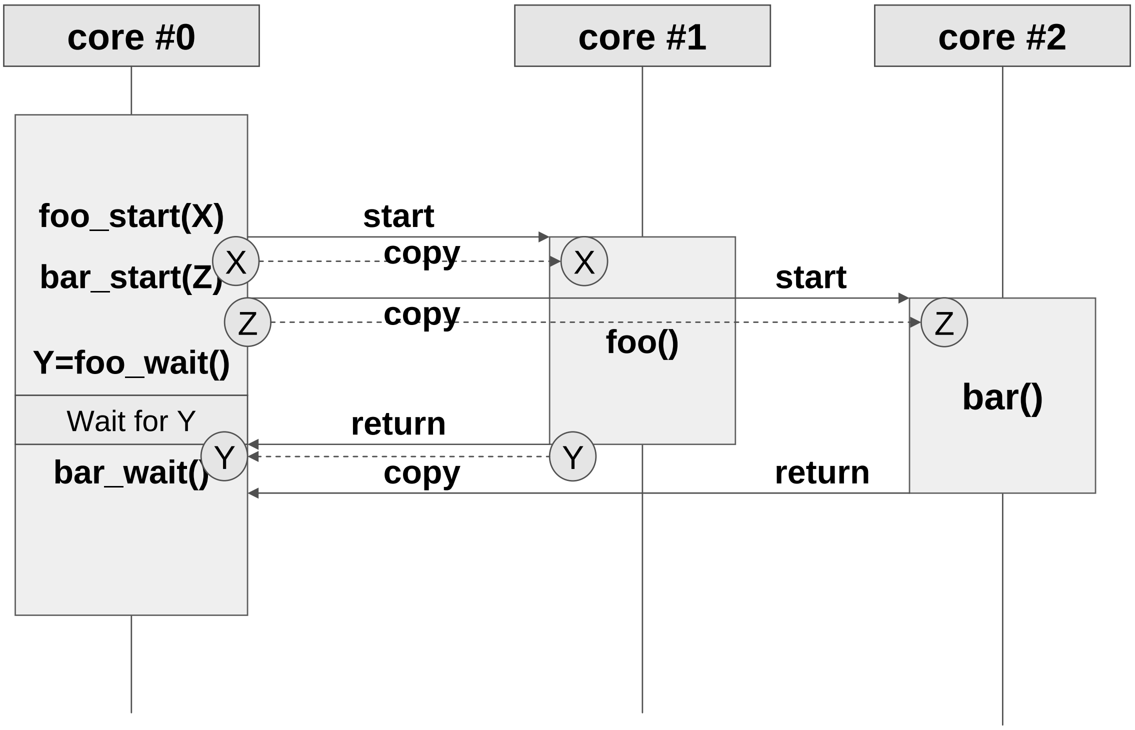
Frontend

Backend

Dual core

Quad core

Sequence chart of ARPC programming



core #0

core #1

core #2

foo_start(X)

bar_start(Z)

Y=foo_wait()

Wait for Y

bar_wait()

start

copy

start

copy

foo()

bar()

return

copy

return

X

X

Z

Z

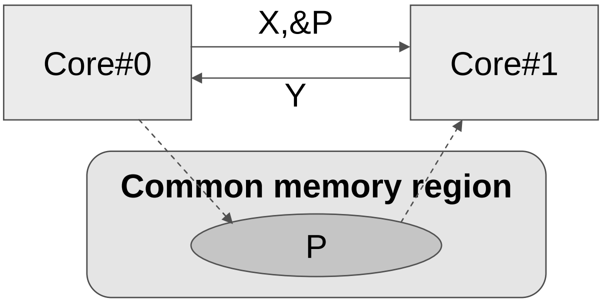
Y

Y

Delivery of pointer data

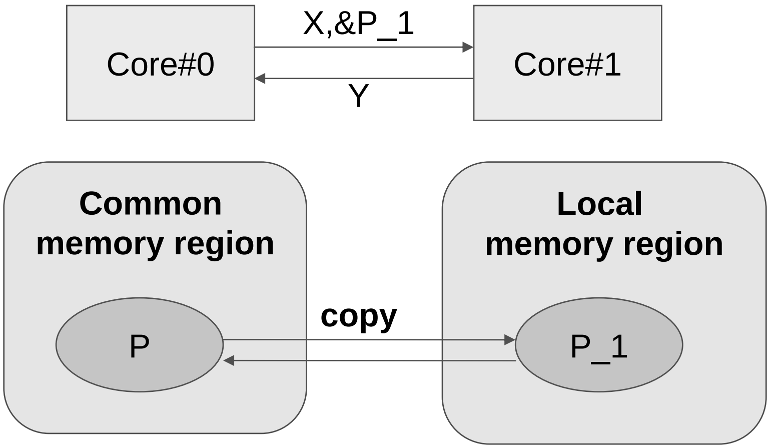
Common memory region

```
foo_start(&handle, X, &P);

…          } Inhibit  access to P

Y=foo_wait(&handle);
```

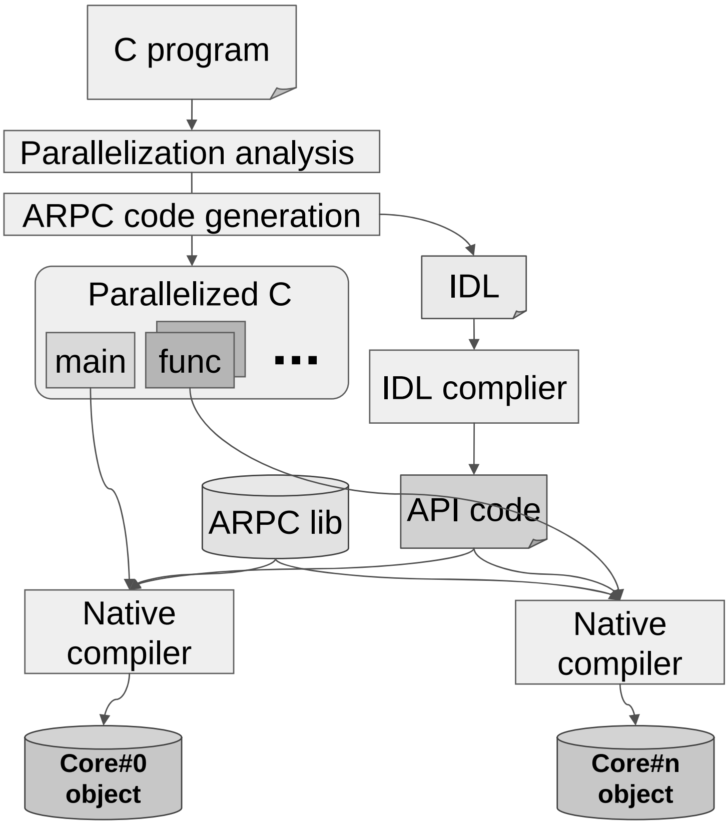


Local memory region

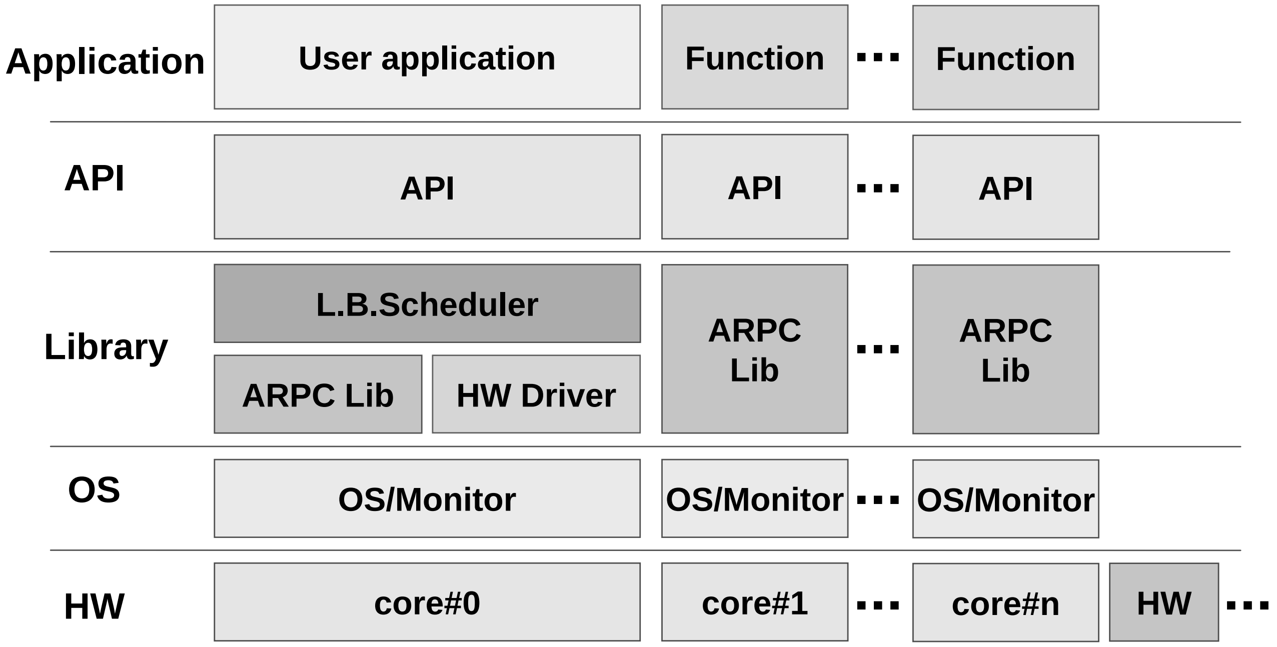
```
move_start(&handle1,
           &P_1, &P, size);
move_wait(&handle1);
foo_start(&handle2, X, &P_1);

…

Y=foo_wait(&handle2);
move_start(&handle1,
           &P, &P_1, size);
move_wait(&handle1);
```



Program development flow



System block diagram



Application: User application | Function ··· Function

API: API | API ··· API

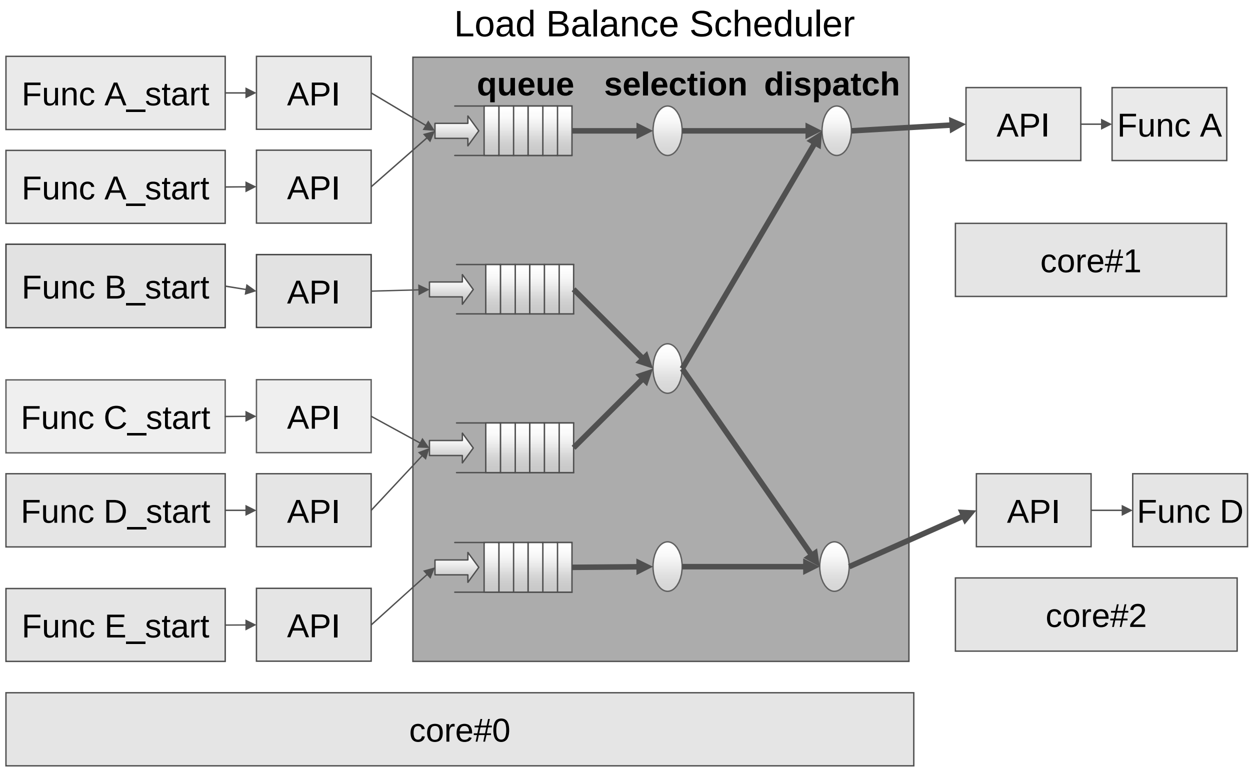
Library: L.B.Scheduler, ARPC Lib, HW Driver | ARPC Lib ··· ARPC Lib

OS: OS/Monitor | OS/Monitor ··· OS/Monitor

HW: core#0 | core#1 ··· core#n | HW ···

IPC: Inter-Processor Communication

Communication flow



Load Balance Scheduler

queue selection dispatch

Func A_start → API

Func A_start → API

Func B_start → API

Func C_start → API

Func D_start → API

Func E_start → API

core#0

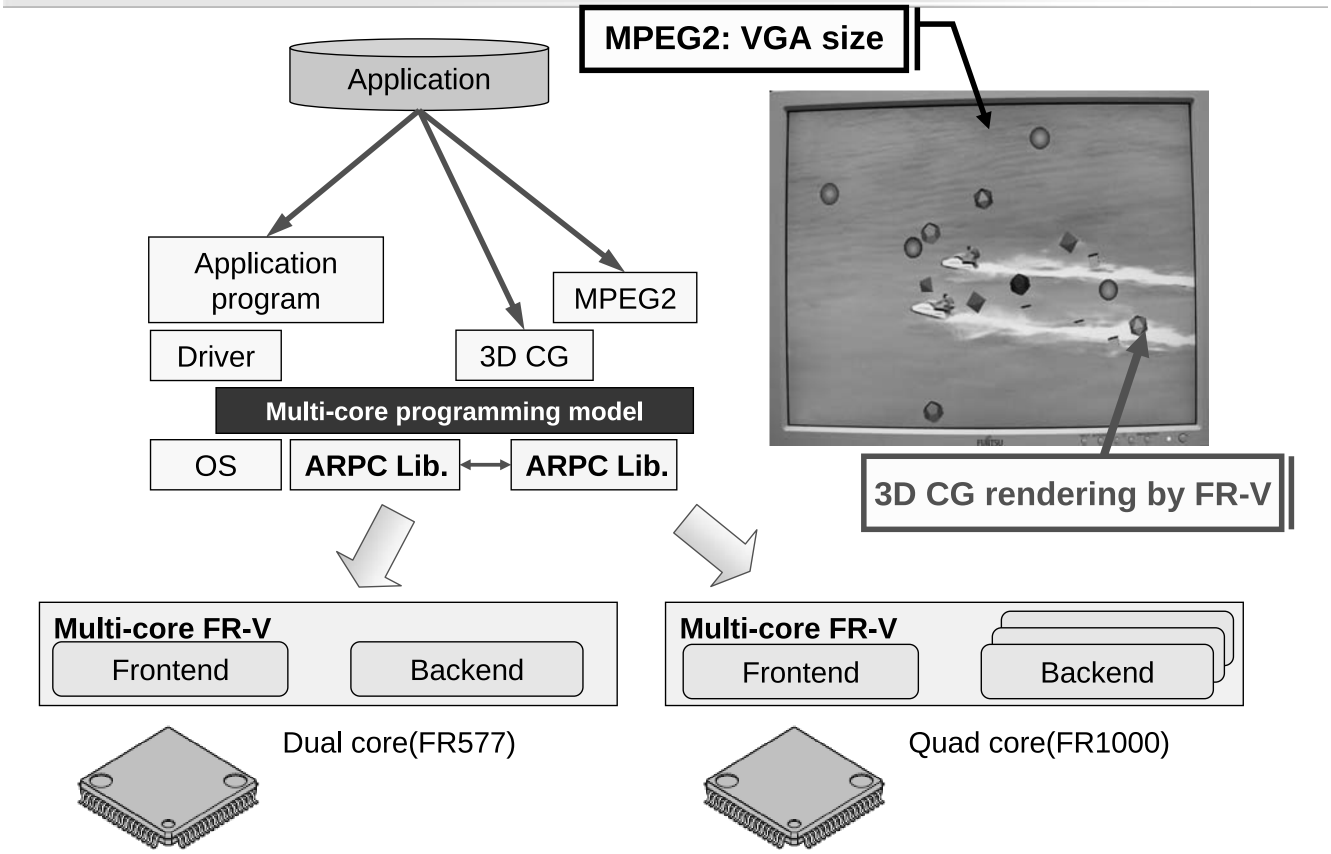
API → Func A

core#1

API → Func D

core#2

Examples of ARPC program



Summary

- We propose an ARPC based programming environment with load balancing technology as an embedded multi-core processor programming technology

  - We show API description and programming behavior.

  - We show demonstration in case of FR-V processor.

ARPC based multi-core programming and Load balancing technology implement easy migration from single to embedded multi-core

FUJITSU

THE POSSIBILITIES ARE INFINITE