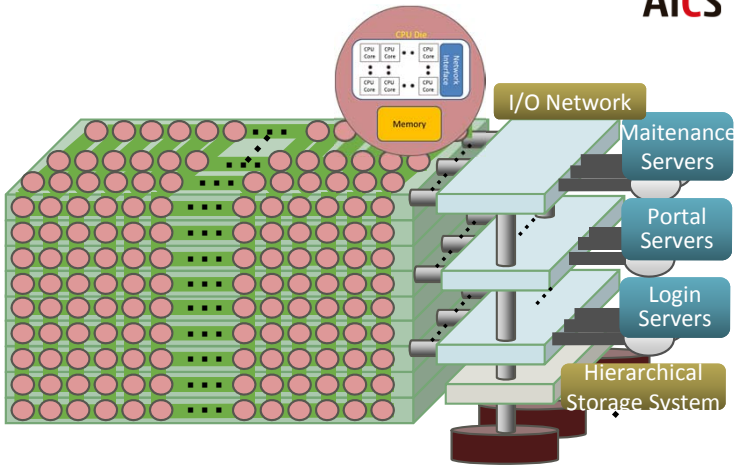
FLAGSHIP 2020 Project:
Development of “Post-K” and ARM SVE

Mitsuhisa Sato Team Leader of Architecture Development Team

FLAGSHIP 2020 project
RIKEN Advance Institute of Computational Science (AICS)

FLAGSHIP2020 Project



Missions

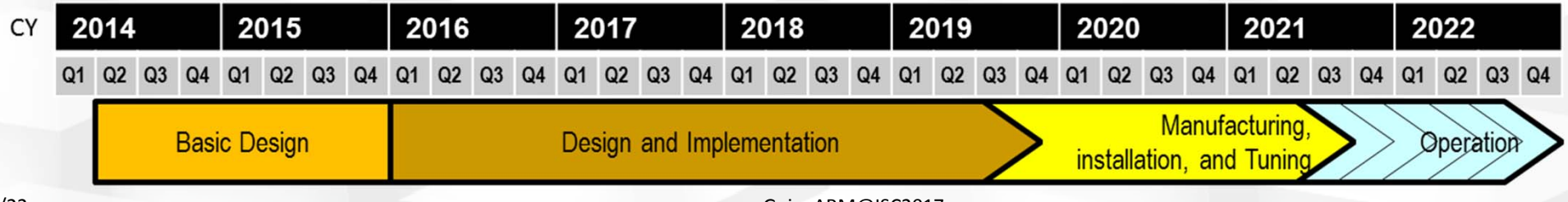
- Building the Japanese national flagship supercomputer, post K, and
- Developing wide range of HPC applications, running on post K, in order to solve social and science issues in Japan

Project organization

- Post K Computer development
  - RIKEN AICS is in charge of development
  - Fujitsu is vendor partner.
  - International collaborations: DOE, JLESC, ..
- Applications
  - The government selected 9 social & scientific priority issues and their R&D organizations.

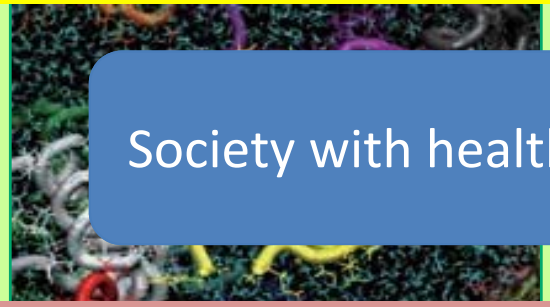
Status and Update

- "Basic Design" was finalized and now in "Design and Implementation" phase.
- We have decided to choose ARM v8 with SVE as ISA for post-K manycore processor.
- We are working on detail evaluation by simulators and compilers



Target science: 9 Priority Issues

①Innovative Drug Discovery



RIKEN Quant. Biology Center

②Personalized and Preventive Medicine

Inst. Medical Science, U. Tokyo

Society with health and longevity

③Hazard and Disaster induced by Earthquake and Tsunami

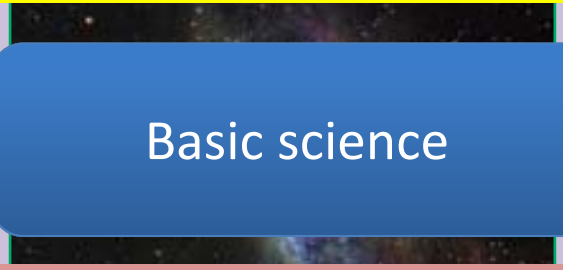
Disaster prevention and global climate

⑧ Innovative Design and Production Processes for the Manufacturing Industry in the Near Future

C

Industrial competitiveness

⑨Fundamental Laws and Evolution of the Universe



Cent. for Comp. Science, U. Tsukuba

Basic science

④Environmental Predictions with Observational Big Data

Center for Earth Info., JAMSTEC

⑦New Functional Devices and High-Performance

Inst. For Solid State Phys., U. Tokyo

⑥Innovative Clean Energy Systems

Grad. Sch. Engineering, U. Tokyo

Energy issues

⑤High-Efficiency Energy Creation, Conversion/Storage and Use

Inst. Molecular Science, NINS

Target science: Exploratory Issues

Interactive Models of Socio-Economic Phenomena and their Applications

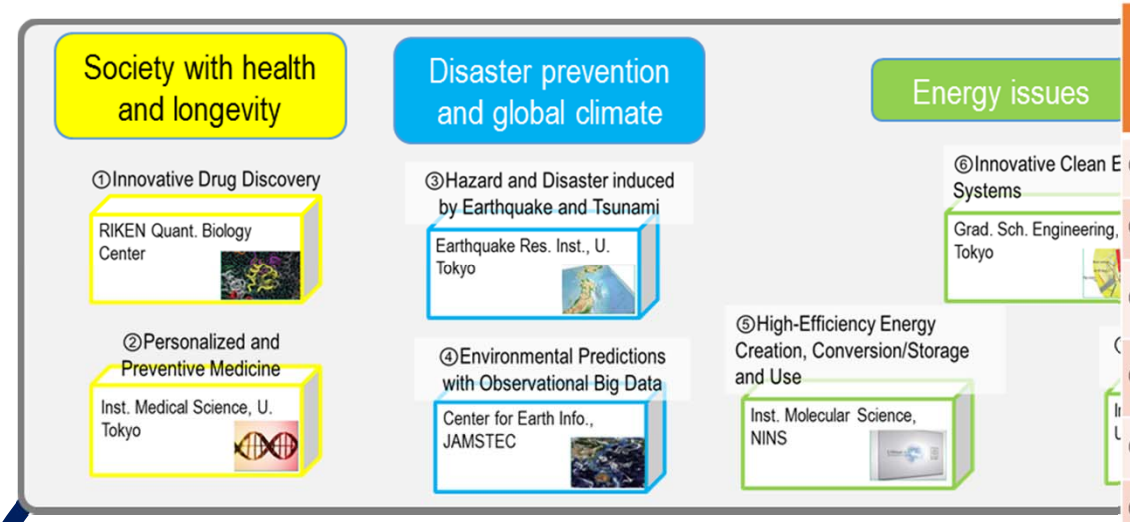
Frontiers of Basic Science - challenge to extremes -

Formation of exo-planets (second Earth) and Environmental Changes of Solar Planets

Mechanisms of Neural Circuits for Human Thoughts and Artificial Intelligence

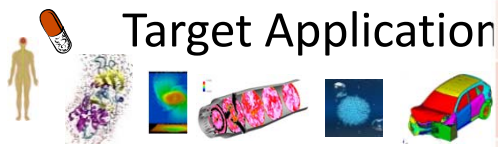
Projects (more than 10 teams) were selected in this Jun 2016

Co-design



	Target Application	
	Program	**Brief description**
①	GENESIS	MD for proteins
②	Genomon	Genome processing (Genome alignment)
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)
⑤	NTChem	molecular electronic (structure calculation)
⑥	FFB	Large Eddy Simulation (unstructured grid)
⑦	RSDFT	an ab-initio program (density functional theory)
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)

Target Application



Architectural Parameters

- #SIMD, SIMD length, #core, #NUMA node
- cache (size and bandwidth)
- memory technologies
- specialized hardware
- Interconnect
- I/O network

FUJITSU

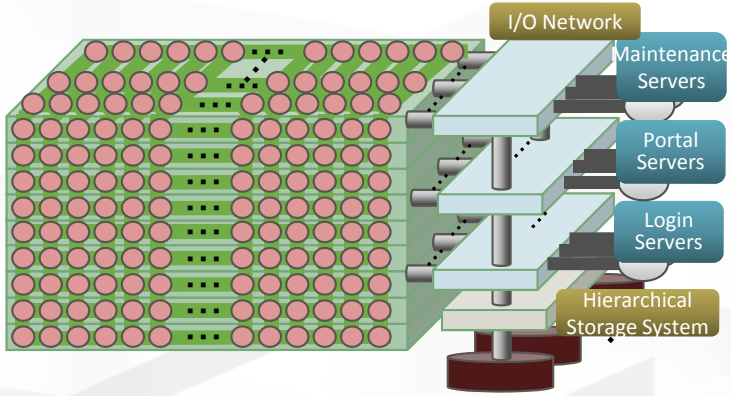
RIKEN AICS

Target Applications' Characteristics

	Target Application		
	Program	Brief description	Co-design
①	GENESIS	MD for proteins	Collective comm. (all-to-all), Floating point perf (FPP)
②	Genomon	Genome processing (Genome alignment)	File I/O, Integer Perf.
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)	Comm., Memory bandwidth
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)	Comm., Memory bandwidth, File I/O, SIMD
⑤	NTChem	molecular electronic (structure calculation)	Collective comm. (all-to-all, allreduce), FPP, SIMD,
⑥	FFB	Large Eddy Simulation (unstructured grid)	Comm., Memory bandwidth,
⑦	RSDFT	an ab-initio program (density functional theory)	Collective comm. (bcast), FPP
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)	Comm., Memory bandwidth, SIMD
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)	Comm., Memory bandwidth, Collective comm. (allreduce)

An Overview of post K

- **Hardware**
  - Manycore architecture
  - 6D mesh/torus Interconnect
  - 3-level hierarchical storage system
    - Silicon Disk
    - Magnetic Disk
    - Storage for archive

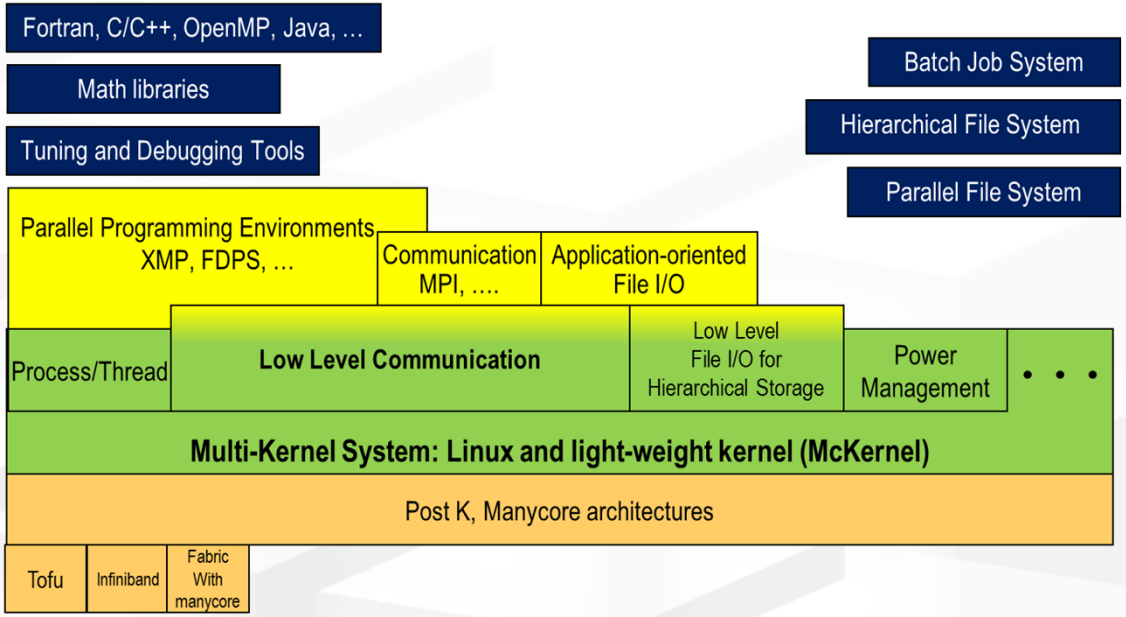


- **System Software**
  - Multi-Kernel: Linux with Light-weight Kernel
  - File I/O middleware for 3-level hierarchical storage system and application
  - Application-oriented file I/O middleware
  - MPI+OpenMP programming environment
  - Highly productive programming language and libraries

MC-kernel: a lightweight Kernel for manycore

XcalableMP PGAS language

FPDS DSL



Post-K: Powered by Fujitsu-designed CPU & Tofu

FUJITSU

- Fujitsu CPU cores support the ARM SVE instruction set architecture
- Post-K Fujitsu CPU cores & Tofu maintain the programming models and provide high application performance
- ARM's standard frameworks (SBSA, etc.) assure compatibility among platforms
- FP16

	Functions & architecture architecture	Post-K	FX100	FX10	K
CPU Core	Instruction set architecture	ARMv8-A	SPARC V9		
	SIMD width	512bit	256bit	128bit	128bit
	Double precision (64bit)	✔	✔	✔	✔
	Single precision (32bit)	✔	✔	✔	✔
	Half precision (16bit)	✔	-	-	-
Interconnect	Tofu interconnect	Enhanced	Tofu2	Tofu	Tofu

Post-K Supports a New Data Type: FP16

FUJITSU

- FP16 is one of the key features for further optimization opportunities regarding hardware, system software, and applications

Challenges

- Power consumption
- Bandwidth
- Cost

Reducing data bits & their movement

- Smaller data types resolve, directly

Modern Processors

GPUs, Xeon Phi and Fujitsu processors will support: new data types

Post-K CPU will support

Packed vectors of:

- Half precision FP16 elements
- 8 & 16 bit fixed-point elements

Applications

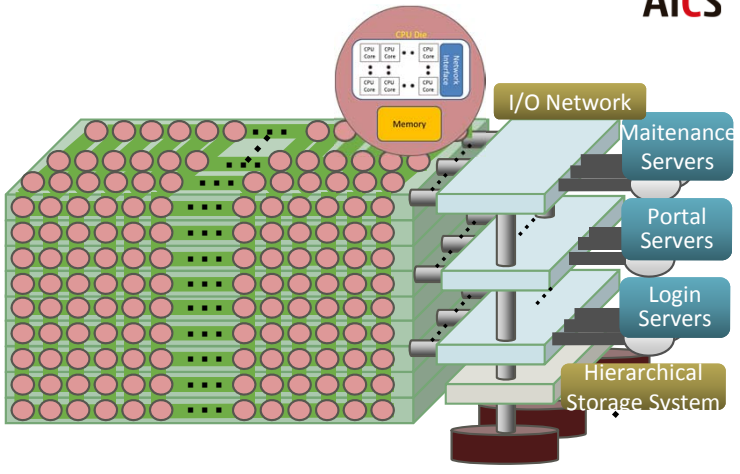
- Existing numerical apps
- Brand-new apps, Incl. deep learning

Mixed precision: Linear Algebra

Relaxed precision: Molecular Dynamics

Half precision: Deep Learning

FLAGSHIP2020 Project



Missions

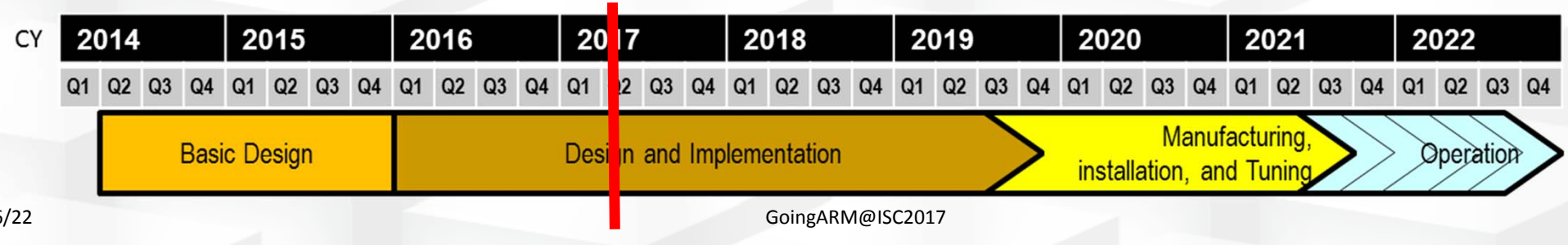
- Building the Japanese national flagship supercomputer, post K, and
- Developing wide range of HPC applications, running on post K, in order to solve social and science issues in Japan

Project organization

- Post K Computer development
  - RIKEN AICS is in charge of development
  - Fujitsu is vendor partner.
  - International collaborations: DOE, JLESC, ..
- Applications
  - The government selected 9 social & scientific priority issues and their R&D organizations.

Status and Update

- "Basic Design" was finalized and now in "Design and Implementation" phase.
- We have decided to choose ARM v8 with SVE as ISA for post-K manycore processor.
- We are working on detail evaluation by simulators and compilers



What we did and what we are doing for ARM SVE

- **Early assessment for ARM SVE spec.**
  - Review of specification of SVE
  - Deployment of ARM-SVE DS-5 tools
- **GEM5 processor simulator for ARM SVE**
  - Deployment and testing of GEM5 Atomic Model developed in ARM
  - Development of GEM5 O3 Model for "Post-K" processor (on-going)
    - Adjustment of parameters and performance with Fujitsu-in-house processor simulator.
- **Evaluation and Testing of compilers for ARM SVE (with Kyoto U.)**
  - ARM compiler for SVE (based on LLVM) (C and C++, F90 by Flang)
  - Fujitsu compiler for SVE (Fortran and C, C++)
  - These compilers are still immature. We give several feedbacks by taking a look at code.
  - Performance evaluation using GEM5 O3 (on-going)
- **Compiler Research on SIMD-vector code-generation for ARM SVE**

Proposal for Explicit SIMD Programming

OpenMP SIMD Directive

- SIMD code generation depends on the compiler
- generated SIMD code may not be optimal

```
#pragma omp declare simd
uchar add_filter(uchar a2, uchar in1, uchar in2) {
  if (a2 > 0) {
    ushort temp = (ushort)in1 + (us
    if (temp > 255) return 255;
    else return (uchar)temp;
  }
  else return in1;
}
```

```
#pragma omp alias simd to(add_filter)
svuint8_t add_filter_acle(svbool_t p, svuint8_t a2,
  svuint8_t in1, svuint8_t in2) {
  svuint8_t zero = svdup_n_u8_x(p, 0);
  svbool_t alpha_mask = svcmpgt_u8(p, a2, zero);
  svuint8_t temp = svand_u8_z(alpha_mask, in2, in2);
  return svqadd_u8(in1, temp);
}
```

ALIAS SIMD directive

- programmer gives the SIMD implementation of the target function
- the directive matches the target scalar function with the SIMD version
- the compiler uses the SIMD version when vectorizing loops

Comparison with ACLE

```
int i = 0;
svbool_t p = svwhilelt_b8_s32(i, N);
svbool_t tp = svptrue_b32();
while (svptest_first(tp, p)) {
  svuint8_t vin1_r = svld1_u8(p, in1_r+i);
  svuint8_t vout_r = add_filter_acle(p,vin2_a,vin1_r,vin2_r);
  svuint8_t vout_g = add_filter_acle(p,vin2_a,vin1_g,vin2_g);
  svuint8_t vout_b = add_filter_acle(p,vin2_a,vin1_b,vin2_b);
  svst1_u8(p, out_r+i, vout_r);
  i += svcntb();
  p = svwhilelt_b8_s32(i, N);
}
```

Vectorizing Loop in ACLE

```
svuint8_t add_filter_acle(svbool_t p, svuint8_t a2,
  svuint8_t in1, svuint8_t in2) {
  svuint8_t zero = svdup_n_u8_x(p, 0); . . .
```

```
#pragma omp simd
for (int i = 0; i < N; i++) {
  out_r[i] = add_filter(in2_a[i], in1_r[i], in2_r[i]);
  out_g[i] = add_filter(in2_a[i], in1_g[i], in2_g[i]);
  out_b[i] = add_filter(in2_a[i], in1_b[i],
}
```

OpenMP +
Explicit SIMD implementation

```
#pragma omp alias simd to(add_filter)
svuint8_t add_filter_acle(svbool_t p, svuint8_t a2,
  svuint8_t in1, svuint8_t in2) {
  svuint8_t zero = svdup_n_u8_x(p, 0); . . .
```

Concluding Remarks

- **We believe that ARM SVE will deliver high-performance and flexible SIMD-vectorization to our "post-K" manycore processor.**

- **We think establishment of "eco-system" for ARM SVE in high-end HPC area very important, and are willing to do collaborations with partners who are interested in ARM SVE, as will as ARMv8.**
  - By sharing the experience of compilers and simulators