

Research and Development of Readout System for New Wide-Field CCD Camera of Subaru Telescope

Hironao Miyatake
Department of Physics, University of Tokyo

January 2009

Abstract

The Hyper Suprime-Cam (HSC) is the next generation wide-field camera proposed for the 8.2-m diameter Subaru Telescope operated by the National Astronomical Observatory of Japan. One of the main purposes of the HSC is to put a new constraint on the nature of dark energy through weak lensing survey.

The HSC employs ~ 110 2kx4k Charge Coupled Devices (CCDs) to cover a 1.5 degree diameter field of view. The readout electronics is required to send ~ 2.1 Gbytes of image data to downstream within 10 seconds. The noise from the readout system should be lower than that from the CCD, which is expected to be $\sim 4e^-$. The readout system should be physically small and lightweight in order for it to be able to be mounted inside the camera unit.

We have developed a prototype readout module for the Back-End Electronics (BEE) of the readout system. The BEE is a digital component of the readout system, which provides an interface between an analog front end and a data acquisition system (DAQ system). Two Gigabit Ethernet links are employed for data transfer between the BEE and DAQ system in order to satisfy the data transfer rate requirement. The size of our BEE module is 149mm x 79mm x 12mm. We measured the performance of the BEE module. The data rate from the BEE module to the DAQ system reached full Gigabit Ethernet bandwidth and the noise induced by the BEE module is estimated to be $\sim 0.05e^-$.

We have built the prototype readout system using the BEE module and the analog front end of the existing wide-field camera of the Subaru Telescope. Using this system, we predicted the behavior of the readout system for the HSC. This demonstrates the feasibility of the HSC readout system using our BEE module.

Contents

1	Introduction	6
2	Dark Energy and the Hyper Suprime-Cam	8
2.1	Basics of Cosmology and Dark Energy	8
2.2	Weak Lensing Survey	10
2.2.1	Basic of Weak Lensing	10
2.2.2	Weak Lensing as a Probe of Dark Energy	12
2.3	Subaru Telescope and Hyper Suprime-Cam	14
2.3.1	Subaru Telescope	15
2.3.2	Overview of Hyper Suprime-Cam	16
2.3.3	Comparison with other projects	19
3	Overview of the Readout System for the Hyper Suprime-Cam	20
3.1	Design Concept	20
3.2	Front End Electronics	22
3.3	Back End Electronics	23
4	R & D of the Readout Module for the Back End Electronics	26
4.1	Requirement	26
4.2	Overview of the readout module	27
4.2.1	Design	27
4.2.2	Operation	29
4.2.3	Communication with the DAQ system	29
4.3	FPGA implementation	32
4.3.1	Clock Management	32
4.3.2	SiTCP	34
4.3.3	Readout Controller	34
4.3.4	Frame Memory Controller	41
4.4	Performance of Readout Module for Back-End Electronics	53
4.4.1	Throughput	53
4.4.2	Power Consumption	57
4.4.3	Noise Estimation	58

5	Prototype Readout System for Hyper Suprime-Cam	62
5.1	Prototype Readout System	62
5.2	Performance of Prototype Readout System	65
5.2.1	Time Dependence of an output of the FEE	65
5.2.2	Temperature Drift of CDS's gain	66
5.2.3	Noise at fixed temperatures	70
6	Conclusion	73
6.1	Summary	73
6.2	Future Prospect	73
A	Details of UDP Communication with GESiCA	75
A.1	RBCP	75
A.2	RBCP Communication with GESiCA	77

List of Figures

2.1	Light Deflection by Gravitational Lens	11
2.2	Effect of Weak Lensing	13
2.3	Cosmic Shear Power Spectra Divided in Three Redshift Slices	14
2.4	Position of Prime Focus Camera	15
2.5	Structure of the Hyper Suprime-Cam	16
2.6	HSC CCD Alignment	17
2.7	FDCCD	18
3.1	A schematic of the readout system for the HSC.	21
3.2	Correlated Double Sampling of CCD Signal. In a typical CCD signal, the reset part (corresponding to $0e^-$) comes first, and data part (corresponding to the number of electrons which pixel acquired) next. The issue is that there is noise that adds the bias to the whole signal of one pixel, which is called kTC noise. To cancel this noise, the reset part and the data part are integrated respectively and their difference is output. This process is called correlated double sampling.	22
3.3	FEE Boards Alignment. The FEE boards are located under the cold plate in the cryostat	23
3.4	Schematics of a planned BEE system	24
4.1	Photograph of prototype readout module (GESiCA) (front view)	28
4.2	Photograph of prototype readout module (GESiCA) (top view)	28
4.3	A block diagram of GESiCA	31
4.4	Image Data Header Specification. The size of the header is 32 bytes.	31
4.5	Clock Distribution in FPGA	33
4.6	I/O of SiTCP	35
4.7	Architecture of the Readout Controller.	39
4.8	I/O of the Frame Memory Controller	42
4.9	Timing Chart of the Signal from the FEE	43
4.10	Schematic of Frame Memory Controller	44
4.11	Architecture of the DDR2 SDRAM controller	46
4.12	State Transition Diagram of the DDR2 SDRAM controller	47
4.13	Timing Chart of Sequential Write	48
4.14	Timing Chart of Read	49

4.15	Architecture of the Arbiter	50
4.16	An Example of Arbitration Timing	52
4.17	Setup for Throughput Measurement	54
4.18	Picture of Test System Used in Throughput Measurement	54
4.19	Throughput Measurement: Throughput Time Dependence	55
4.20	Throughput Measurement: Throughput vs Input Rate	56
4.21	Setup of Power Consumption Measurement	57
4.22	Power Consumption Measurement	58
4.23	Setup of the Jitter Measurement	59
4.24	Jitter Measurement of Readout Clock. Y-axis is the jitter in seconds.	61
4.25	Jitter Measurement of Readout Clock. Data is same as Fig.4.24, but Y-axis is the relative accuracy in ppm.	61
5.1	Picture of the Prototype Readout System	63
5.2	Schematic of Prototype Readout System	63
5.3	Interface Board for GESiCA	64
5.4	Output Time Dependence	66
5.5	Gain of the CDS	67
5.6	Temperature Drift of the Gain of the CDS	68
5.7	Noise of the Output (20 degree C).	71
5.8	Noise of the Output (0 degree C).	71
A.1	Schematic of communication over RBCP	75
A.2	RBCP Write Sequence	76
A.3	RBCP Read Sequence	77

List of Tables

2.1	FDCCD Specification. All parameters are measured in -100 degree Celsius .	18
2.2	Ongoing Projects for Probing Dark Energy	19
3.1	Electronics Requirements	20
4.1	Pin assignment of the CMC connector of GESiCA. PIN0-3/POUT0-3 are general purpose inputs/outputs. RESET is used for master reset. ACTIVE is used for return from sleep. SYNC0-1 are used for synchronization of readout modules. UART0-1 are used for serial communication. The signals mentioned above will be used in the future. FG_SCLK and FG_LOAD are the control signal from the FEE. FG0-15 is the 16-bit digital image data from the FEE. CP0-31 is the 32-bit readout clock for the CCDs and the FEE.	30
4.2	User Interface Signals of SiTCP	35
4.3	Structure of the Memory of the Pattern Generator	40
4.4	Structure of the Memory in the Pattern Sequencer	40
4.5	DDR2 SDRAM Initialization Settings	48
5.1	Gain of the CDS from the Fit	68
5.2	Relative Temperature Drift	69
5.3	Result of Fit	72
A.1	Packet Format of RBCP	76
A.2	Address Map for GESiCA Control	80
A.3	Address Map for Each Octet of setExpSize Address	80
A.4	Address Map for Each Octet of PtRAM Address	80
A.5	Address Map for Each Octet of PrRAM Address	80

Chapter 1

Introduction

The drastic advancements in astronomical observation technology in the last decade has cast a profound issue straddling the line between particle physics and cosmology: dark energy, an unidentified energy that fills the entire space and increases the rate of expansion of the universe. In 1998, the observation of type Ia supernovae suggested that the expansion of the universe is accelerating[1]. The measurement of the cosmic microwave background (CMB) by the Wilkinson Microwave Anisotropy Probe (WMAP) has made a strong confirmation of the existence of dark energy[3]. According to the result from WMAP, the energy of the universe is made up of 72 % dark energy, 23 % dark matter, and 5 % of matter we already know (baryons). It is amazing that the main part of the universe is filled with such a peculiar energy.

On a superficial level, dark energy seems to be cosmological constant which Einstein introduced about nine decades ago . However, it constitutes a contradiction with particle physics; present particle physics cannot explain what field dark energy is. For instance, the natural value of the cosmological constant predicted by particle physics is $\Lambda = c^2/(\hbar G) \sim 5.2 \times 10^{93} \text{ g/cm}^3$, which results in $\Omega_{de} \sim 10^{121}$. On the other hand , as we mentioned above, observations report $\Omega_{de} \sim 0.72$. There is an order of 10^{121} of discrepancy between the prediction and the observation, making it one of the worst predictions in history. In addition, dark energy may suggest a limitation of general relativity; general relativity may be no longer applicable on extremely large scales. There is a possibility that dark energy will fundamentally change our view of the world.

To obtain a hint to identify dark energy, it is important to investigate the nature of dark energy such as its time evolution. One of the most efficient methods to survey dark energy is to map a 3D distribution of mass using weak lensing. Many cosmological observations including galaxy surveys revealed that the universe has a vast hierarchy which results in concavity and convexity of space structure. According to Einstein's general relativity, the concavity and convexity of space curves light from distant galaxies. As a result, observers observe systematically distorted images of galaxies. This phenomenon is called cosmological weak lensing. Gravity caused by dark matter is crucial for forming the hierarchical structure of the universe. Dark energy also influences the hierarchy through the expansion of the universe. Therefore observations of the cosmological weak lensing will

help us to understand the nature of dark energy. Measurements of weak lensing have an advantage in smaller systematic error compared with other measurements since there is no assumption of composition or state of matter. In addition, combining other independent methods (CMB, supernovae, and baryon acoustic oscillation) will give stringent constraints on the nature of dark energy.

It is crucial to take extremely high-resolution images of galaxies since cosmic weak lensing effect is very small ($\sim 1\%$ in ellipticity). The existing prime focus camera mounted on Subaru Telescope, the Suprime-Cam, has an advantage in this regard since the Subaru Telescope has superb image resolution. However, the field of view of the Suprime-Cam is not large enough to measure the influence for weak lensing caused by dark energy. Therefore we are developing the new wide-field camera called Hyper Suprime-Cam (HSC)[9]. The field of view of the HSC is 1.5 degrees in diameter which is ~ 7 times larger than that of Suprime-Cam. Surveying 1000 square degrees of the sky (approximately one fortieth of whole sky), we aim to statistically impose stringent limitation on the nature of dark energy.

The HSC employs ~ 110 CCDs. The size of one exposure is ~ 2 Gbytes. The readout system for the HSC processes this large amount of data, which is then transferred to the data acquisition (DAQ) system. The requirement for readout time is that the readout system should send the whole image data of one exposure to the DAQ system within 10 seconds. The noise from the readout system should be smaller than that from the CCD ($\sim 4e^-$). The readout system should be small and light, and its power consumption should be low, because the readout system will be integrated in the camera unit.

Physical motivation and introduction to the HSC will be discussed in Chapter 2, and the overview of the readout system for the HSC will be discussed in Chapter 3. In Chapter 4, the main part of this thesis, we will describe the research and development of the readout module for the digital part of the readout system. In this chapter, the specification, architecture, and evaluation of this module will be discussed. We have built the prototype readout system which consists of the readout module we developed and the analog part of the Suprime-Cam. The evaluation of this system will be described in Chapter 5. We summarize our study in Chapter 6.

Chapter 2

Dark Energy and the Hyper Suprime-Cam

2.1 Basics of Cosmology and Dark Energy

Einstein's equation is written as

$$G_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}, \quad (2.1)$$

where $G_{\mu\nu}$ is the Einstein tensor, Λ is the cosmological constant, $g_{\mu\nu}$ is the metric tensor, G is the gravitational constant, c is the speed of light, and $T_{\mu\nu}$ is the stress-energy tensor. The left-hand side of equation (2.1) represents “geometry of space-time”, while the right-hand side represents “distribution of matter field”. Although they have apparently no relation to each other, Einstein's equation relates them: once one of them is determined, the other is automatically determined. The cosmological constant was originally introduced by Einstein in order to allow static cosmological solutions. The cosmological constant can be interpreted as a part of the stress-energy tensor which is inherent in space and independent of matter, when it is moved to the right-hand side;

$$G_{\mu\nu} = \frac{8\pi G}{c^4} (T_{\mu\nu} - \frac{c^4}{8\pi G} \Lambda g_{\mu\nu}). \quad (2.2)$$

If there was no matter in the universe, that is, $T_{\mu\nu} = 0$, the energy resulting from Λ would exist. In this sense, the second term of the right-hand side of the equation (2.2) is called “vacuum energy.”

Let us postulate that the universe is *homogeneous* and *isotropic*. This postulate is called the cosmological principle. When the universe is isotropic, we can write an actual distance as

$$X(t) = a(t)x, \quad (2.3)$$

$a(t)$ is called the scale factor whose present value is one. While $a(t)$ changes with time, x is independent of time. x is called *comoving distance*. Since we have good evidence that the universe is expanding, $a(t)$ was smaller than one at earlier times. We define

comoving coordinate as the coordinate system where comoving distance is defined. Under the cosmological principle, the metric can be represented as

$$ds^2 = -dt^2 + a^2(t) \left[\frac{dr^2}{1 - Kr^2} + r^2(d\theta + \sin^2\theta d\phi^2) \right], \quad (2.4)$$

where K is a curvature. r, θ, ϕ is the comoving coordinate of the spherical coordinate system. This metric is called Robertson-Walker metric.

The cosmological principle enable us to assume that the universe is filled with the a homogeneous perfect fluid which can be written as

$$T_{\mu\nu} = (\rho c^2 + P)u_\mu v_\nu + P g_{\mu\nu} \quad (2.5)$$

in the form of the stress-energy tensor. Here ρ is the energy density, P is the pressure, and u_μ, v_ν is the four-vector velocity.

Under the cosmological principle which corresponds to equations (2.4) and (2.5), Einstein's equation becomes

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G\rho}{3} - \frac{Kc^2}{a^2} + \frac{\Lambda c^2}{3}, \quad (2.6)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3c^2}(\rho c^2 + 3P) + \frac{\Lambda c^2}{3}, \quad (2.7)$$

which is called Freedmann equations.

Let us assume that the relation between the energy density ρ and P has the form of

$$P = w\rho c^2. \quad (2.8)$$

This equation is called the equation of state. $w = 0$ for matter (non-relativistic particle) and $w = 1/3$ for radiation (relativistic particle). Therefore, due to equation (2.7), it turns out that matter and radiation make a contribution to the deacceleration of the universe's expansion ($\ddot{a} < 0$). Vacuum energy makes a contribution to the acceleration of the expansion ($\ddot{a} > 0$). As for vacuum energy equation of state, from equations (2.2) and (2.5) we can obtain

$$\rho_v = \frac{\Lambda c^2}{8\pi G}, P_v = -\frac{\Lambda c^4}{8\pi G}, \quad (2.9)$$

which results in $w = -1$. Generally, the energy fluid which has $w < -1/3$ accelerates expansion of the universe. We call such energy "dark energy". In this sense, vacuum energy can be interpreted as a special form of dark energy. w may change with time. From now on, we generalize the vacuum energy as dark energy and write ρ_v and P_v as ρ_{de} and P_{de} which have the relation of $P_{de} = w\rho_{de}c^2$

Using the Hubble parameter $H = \dot{a}/a$ and $\Omega = \rho/\rho_{cr,0}$ where the present critical density $\rho_{cr,0} = \frac{3H_0^2}{8\pi G}$ and setting $t = t_0$ (present), equation (2.6) becomes

$$1 = \Omega_{r,0} + \Omega_{m,0} + \Omega_{de,0} + \Omega_{K,0}, \quad (2.10)$$

where $\Omega_{K,0} = -K_0 c^2 / a^2 H_0^2$. Thus energy of the universe is filled with radiation, matter, dark energy, and curvature.

Recent cosmological observations, including the observations of type Ia supernovae[1][2] and cosmic microwave background(CMB), strongly support the Λ CDM model. Λ stands for the cosmological constant (dark energy) that allows for the current accelerating expansion of the universe. CDM stands for cold dark matter; the matter that seldom interacts with others (i.e. dark matter is not baryons) and was cooled (i.e. its velocity is non-relativistic) at the epoch of radiation-matter equality. Cold dark matter is necessary to realize the present structure of the universe. According to the Λ CDM model, $\Omega_{r,0}$ in equation (2.10) = 0.

The cosmological parameters with 68% CL, derived from the Wilkinson Microwave Anisotropy Probe (WMAP) data combined with the type Ia supernova (SN) and baryon acoustic oscillation (BAO) under the Λ CDM model, are [3]: $\Omega_{b,0} = 0.0456 \pm 0.0015$, $\Omega_{dm,0} = 0.228 \pm 0.013$, $\Omega_{de,0} = 0.726 \pm 0.015$, where indices b and dm mean the baryon and the dark matter respectively. In addition, the limitations $-0.14 < 1 + w < 0.12$, $-0.0179 < \Omega_{K,0} < 0.0081$ (95% CL) were obtained, where w is assumed to be constant. These results are consistent with equation 2.10, which supports inflation theory.

Weak lensing surveys are expected to tighten the limitation of the cosmological parameters described above, because they are an observation independent of the other methods. In the next section, we discuss the efficiency of weak lensing surveys.

2.2 Weak Lensing Survey

According to general relativity, mass structure induces distortion of space-time, which results in the deflection of light which go through space-time. Therefore observers observe distorted galaxy images. This is called weak lensing. The distortion is affected not only by matter but also dark energy, since dark energy influences the evolution of the universe through its “negative pressure.” In this section we discuss basic of weak lensing and then how weak lensing is used as a probe of dark energy.

2.2.1 Basic of Weak Lensing

Fig. 2.1 shows the typical system of gravitational lensing, where a mass concentration at angular distance D_d deflects the light from a source at angular distance D_s , that is, the mass concentration behaves as a lens. D_{ds} is angular distance between the lens and the source. Deflected by the deflection angle $\vec{\alpha}(\vec{\theta})$, the true image at the position $\vec{\beta}$ is observed at $\vec{\theta}$. Supposing all the angles are sufficiently small, the relation between these angles is written as

$$\vec{\beta} = \vec{\theta} - \frac{D_{ds}}{D_s} \vec{\alpha}(\vec{\theta}) \equiv \vec{\theta} - \vec{\alpha}(\vec{\theta}), \quad (2.11)$$

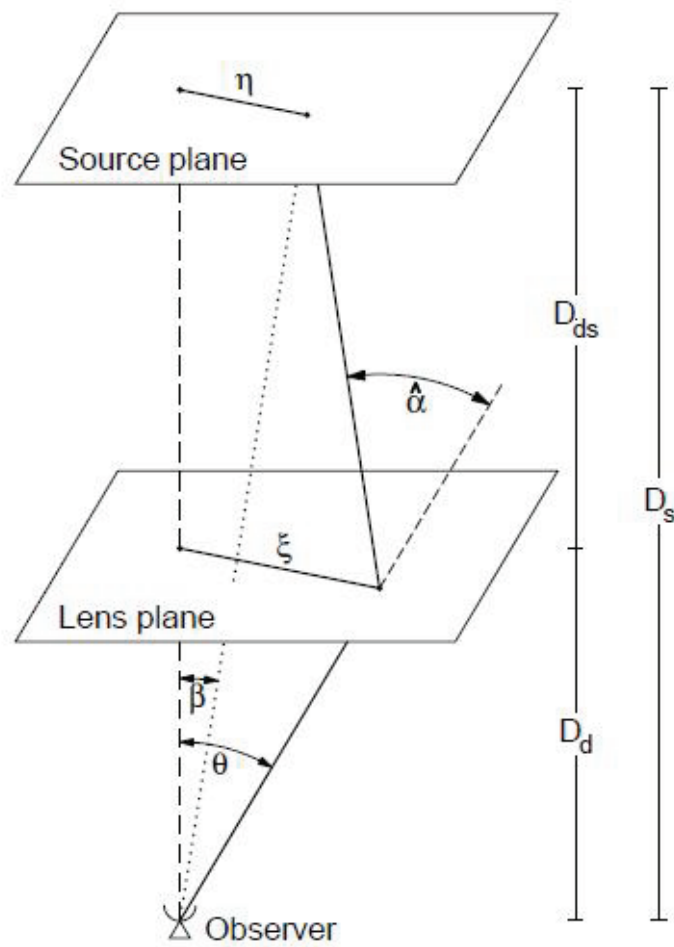


Figure 2.1: Light Deflection by Gravitational Lens

which is called the lens equation. We defined the scaled deflection angle $\vec{\alpha}(\vec{\theta})$ in the last step. General relativity predicts the deflection angle $\vec{\alpha}(\vec{\theta})$ is

$$\vec{\alpha}(\vec{\theta}) = \nabla\psi(\vec{\theta}) = \frac{4G}{c^2} \frac{D_s D_{ds}}{D_s} \int d^2\vec{\theta}' \frac{\vec{\theta} - \vec{\theta}'}{|\vec{\theta} - \vec{\theta}'|^2} \Sigma(D_d \vec{\theta}'), \quad (2.12)$$

where Σ is the 2-dimensional mass density on which mass density of lens objects is projected along the line of sight, represented as $\Sigma(\vec{\theta}) = \int dz \rho(D_d \vec{\theta}, z)$. ψ can be regarded as the bundle potential resulting from the mass structure.

Let us consider the gravitational lensing mapping which can be represented by equations (2.11) and (2.12). The small deviation $(\delta\vec{\beta}, \delta\vec{\theta})$ which moves inside a galaxy image around the center of the galaxy $(\vec{\beta}, \vec{\theta})$ is represented as

$$\delta\vec{\beta} = \frac{\partial\vec{\beta}(\vec{\theta})}{\partial\vec{\theta}} \delta\vec{\theta} \equiv A(\vec{\theta}) \delta\vec{\theta}, \quad (2.13)$$

where

$$A = \left(\delta_{ij} - \frac{\partial^2\psi(\vec{\theta})}{\partial\theta_i \partial\theta_j} \right) = \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix}. \quad (2.14)$$

Here convergence κ and shear $\gamma = \gamma_1 + i\gamma_2 = |\gamma|e^{2i\varphi}$ are

$$\kappa \equiv \frac{1}{2}(\psi_{11} + \psi_{22}), \quad (2.15)$$

$$\gamma_1 \equiv \frac{1}{2}(\psi_{11} - \psi_{22}), \quad (2.16)$$

$$\gamma_2 \equiv \psi_{12} \quad (2.17)$$

For the case of weak lensing, which we define by $\kappa \ll 1$ and $|\gamma| \ll 1$, the transformation matrix A maps the intrinsic image $\vec{\beta}$ into the distorted image $\vec{\theta}$ like Fig. 2.2. We supposed the intrinsic image is a circle. Convergence κ increases the size of the image by $1/(1 - \kappa)$, shown as the left of Fig. 2.2. Shear γ does not change the size but transforms the shape into ellipse (see the right of Fig. 2.2). The induced ellipticity is represented as $e = (a - b)/(a + b) = |\gamma|$ (a, b is length of the major axis and that of the minor axis, respectively) and angle of direction of the major axis is $\phi = (1/2) \arctan(\gamma_1/\gamma_2)$.

2.2.2 Weak Lensing as a Probe of Dark Energy

Weak lensing due to cosmological large-scale structure, including dark energy, causes the distortion of distant galaxies which is called *cosmic shear*. The two-point correlation function of the cosmic shear, $\langle \gamma(\vec{\theta}_1) \gamma(\vec{\theta}_2) \rangle$, is the powerful tool to examine our cosmological model. Since the shear signal due to dark energy is very weak ($\sim 1\%$ in ellipticity), we have to minimize both statistical and systematic error.

We need to extract small cosmic shear signal from galaxies which have inherent ellipticity. Although the ellipticity magnitude of a galaxy is $\sim 30\%$ in average, its direction is likely to be random. This fact implies that we can measure the cosmic shear by averaging

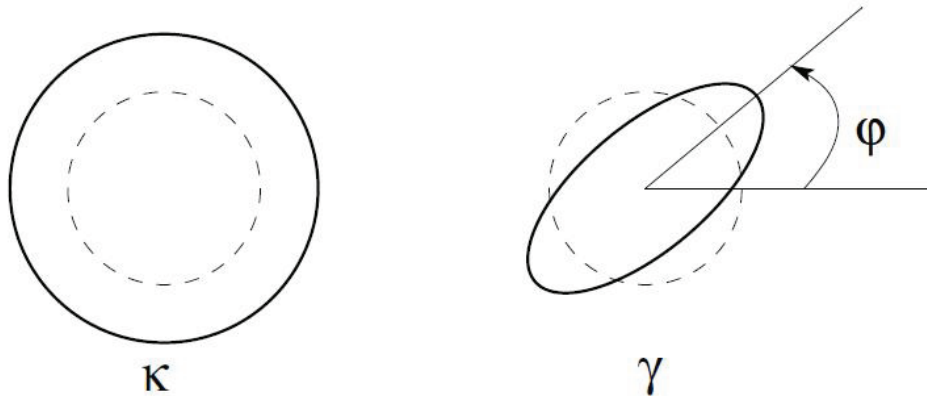


Figure 2.2: Effect of Weak Lensing

the distortions over a sufficiently large number of galaxies and cancelling their intrinsic ellipticity. Thus, the cosmic shear correlation function is an intrinsically statistical quantity. To reduce the statistical error, a large number of galaxies images are required.

Another key to extract the cosmic shear signal precisely is measuring shape of galaxies with a high degree of accuracy from contamination due to atmospheric seeing effect and telescope distortion. To this end, well-resolved and excellent image quality is required. As a measure of resolution, the Shear TEsting Programme (STEP), the project of simulation for cosmic shear measurement, reports image quality with seeing of $\sim 0.6''$ (typical Subaru Telescope seeing) and pixel resolution of $\sim 0.2''$ is good enough for precise measurement of the shear effect[8].

Power spectrum, the Fourier transformed counterpart of the cosmic shear correlation function, is estimated to depend roughly on the cosmological parameters from a perturbative calculation based on the Λ CDM model[7]:

$$P \propto \Omega_{DE}^{-3.5} \sigma_8^{2.9} z_s^{1.6} |w|^{0.31}, \quad (2.18)$$

where σ_8 is the mass dispersion on a scale of $8 h^{-1} \text{Mpc}$ (h is defined as $H_0 = 100h \text{ km s}^{-1} \text{Mpc}^{-1}$), and z_s is the redshift of the source galaxies and the dark energy equation state w is assumed to be constant. Equation (2.18) is the approximation around $z_s = 1$ and multipoles $l = 1000$.

We must consider degeneracies between these cosmological parameters. One way to resolve the degeneracies is binning weak lensing by z_s . This method is called *weak lensing tomography*. The multi-color data set of the weak lensing survey can be used to estimate the distances of galaxies z_s , the so-called photo- z , by utilizing the fact that galaxies display characteristic photometric features depending on the redshift and type of galaxies. Weak lensing tomography is expected to impose tighter restrictions on the nature of dark energy[6].

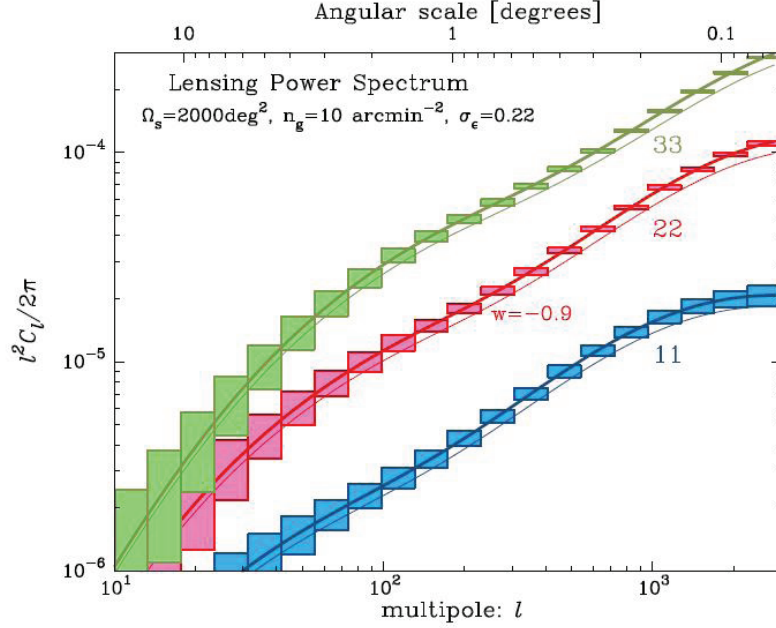


Figure 2.3: Cosmic Shear Power Spectra Divided in Three Redshift Slices

Here we describe an example of applying weak lensing tomography for estimating cosmological parameters having a relation to dark energy. Fig. 2.3 shows the expected cosmic shear power spectra for galaxy distribution divided in three redshifts slices, $0 \leq z_1 \leq 0.6$, $0.6 \leq z_2 \leq 1$, $z_3 \geq 1$. The bold solid curves are power spectra for Λ CDM model, while the thin line curves show the results for a model with $w = -0.9$. The boxes around the bold lines are the expected measurement error for the weak lensing survey that is parametrized by survey area $\Omega_s = 2,000 \text{ deg}^2$, the average number density of galaxies usable for lensing analysis, $\bar{n}_g = 30 \text{ arcmin}^{-2}$, and the rms intrinsic ellipticities, $\sigma_\epsilon = 0.22$. This figure shows the possibility of splitting degeneracies. For example, if σ_8 is fixed by using the WMAP data, we can estimate not only Ω_{de} and (constant) w , but also time dependency of w . These information will give us a significant clue to reveal the identity of dark energy.

2.3 Subaru Telescope and Hyper Suprime-Cam

The requirements for observation to extract the information of dark energy by weak lensing survey are

1. extremely wide-field survey, and
2. well-resolved and undistorted image.

We are developing the next generation wide-field camera satisfying above requirements, which will be mounted on Subaru Telescope. In this section, we discuss the Subaru

Telescope, the new camera and comparison with other survey projects.

2.3.1 Subaru Telescope

Subaru Telescope is located at the 4,200-meter summit of Mauna Kea on the island of Hawaii, which commenced operation in 1999. The Subaru Telescope is operated by National Astronomical Observatory of Japan (NAOJ).

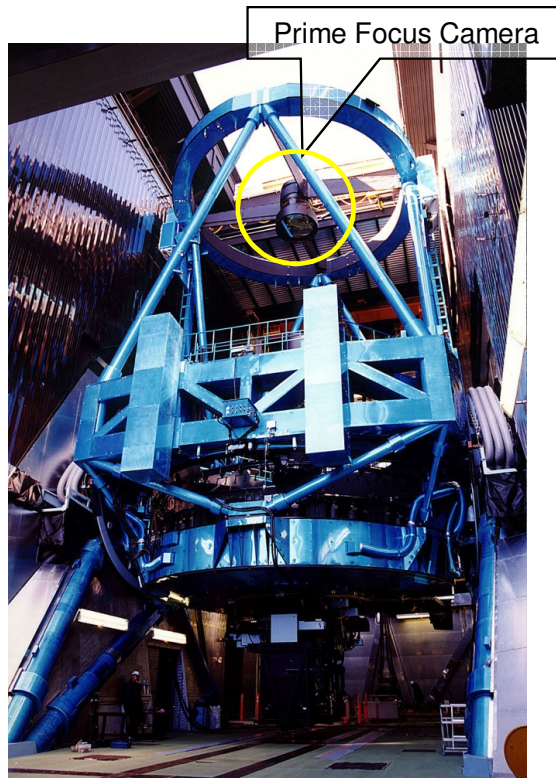


Figure 2.4: Position of Prime Focus Camera

The features of Subaru Telescope are as follows:

- 8.2-m large primary mirror
Subaru Telescope has a single 8.2-m large primary mirror which enables us to survey darker objects and the deeper field.
- prime focus
Generally, the larger primary mirror a telescope has, the smaller field of view (FOV) becomes. However, Subaru Telescope has prime focus where we can set a wide-field camera. This feature is unique among many telescopes in the world.

- high-quality imaging

Subaru Telescope has actuators under its 8.2-m primary mirror to compensate distortion of the mirror. This feature gives us extremely high-quality images.

Subaru Telescope currently mounts Suprime-Cam on its prime focus[10]. The FOV of Suprime-Cam is $0.34 \text{ degree} \times 0.27 \text{ degree}$. The Suprime-Cam has been providing un-vignetted, wide- and deep-field images. In fact, superb images better than 0.6 arc-sec (FWHM) are routinely obtained by Suprime-Cam. Thanks to the Suprime-Cam, dark halos are observed by weak lensing survey[11]. In addition, cosmic shear statistics becomes possible, resulting in rejection of the cosmological model without cosmological constant[12].

2.3.2 Overview of Hyper Suprime-Cam

Although the Suprime-Cam provides high-quality and wide-field images, its FOV is not enough to survey dark energy. We are developing a new wide-field prime focus camera of Subaru Telescope, which is called Hyper Suprime-Cam (HSC), whose FOV is 1.5 degrees in diameter and first light on sky is scheduled on 2011. This wide field enables our survey speed to increase approximately 10 times faster than that of Suprime-cam.

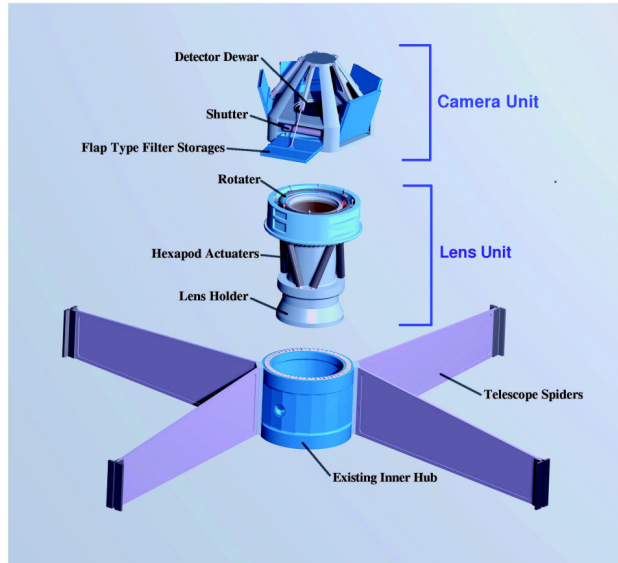


Figure 2.5: Structure of the Hyper Suprime-Cam

Fig. 2.5 shows the structure of the HSC. The structure consists of two units: a lens unit and a camera unit. The lens unit is attached to the telescope from zenith, and the lens barrel containing optics is supported by the hexapod actuators so that the attitude can be controlled. The camera unit is attached to the end of the lens barrel.

Optics

The optical specification is represented as the 80 % encircled energy diameter, D_{80} . In the red band ($\lambda > 600$ nm), where shape measurement of galaxies are usually carried out for weak lensing analysis, D_{80} should be equivalent with that of Suprime-Cam (~ 0.3 arcsec). For the blue region ~ 0.3 arcsec is goal. The optics satisfying this requirement is under development.

CCD

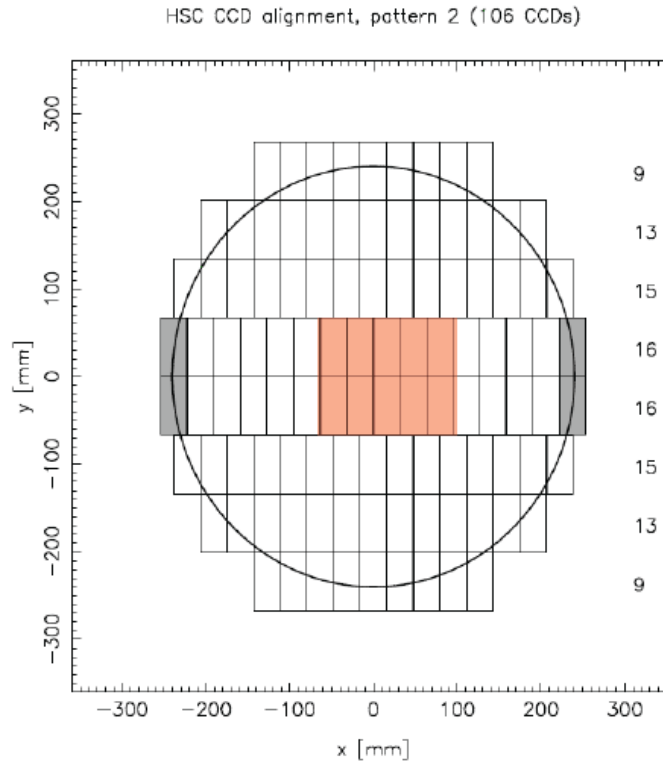


Figure 2.6: HSC CCD Alignment

We use ~ 110 CCDs to cover FOV of 1.5 degree diameter as shown in Fig. 2.6. The orange region represents the area of CCDs corresponding to the Suprime-Cam, while the gray region indicates CCDs used for guide. Fully depleted CCDs (FDCCDs) which was developed in collaboration between NAOJ and Hamamatsu Photonics K.K, are employed (Fig. 2.7). The pixel size of the FDCCD is 2048×4096 ($15 \mu\text{m}$ pixel) which corresponds to 0.17 arcsec in combination with optics and FDCCD has 4 outputs. The important advantage of FDCCD is higher quantum efficiency (QE) in the red band thanks to the thicker depletion layer. FDCCD we developed also has high QE in the blue band due to new backside treatment. FDCCDs are actively cooled down to -100 degree Celsius by cold

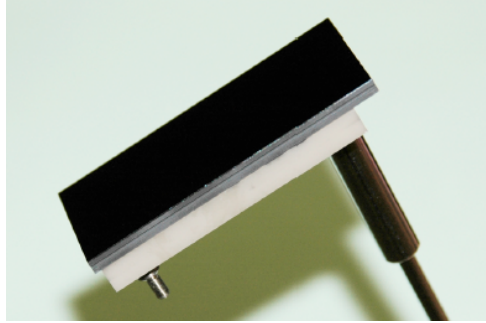


Figure 2.7: FDCCD

plane [16] in order to suppress thermal noise and elicit its performance. The specification of FDCCD is shown in table 2.1. Basic specifications such as charge transfer efficiency (CTE), QE, dark current, and readout noise are all met. Refer to Kamata *et al.*[13] for more details.

Number of Pixels	2048 x 4096
Pixel Size	15 μm
Parallel CTE	0.999995
Serial CTE	0.999995
QE	40% (400nm) 90% (650 nm) 40% (1000 nm)
Dark Current	1.4 $\text{e}^-/\text{hour}/\text{pixel}$
Fullwell	180,000 e^-
Amplifier Responsivity	5 $\mu\text{V}/\text{e}^-$
Readout Noise	4.4 e^- at 150 kHz readout

Table 2.1: FDCCD Specification. All parameters are measured in -100 degree Celsius

Filter

We are planning to employ the following broadband filters:g(400nm - 550nm), r(550nm - 700nm), i(700nm - 850nm), z(850nm - 1000nm), y(950nm - 1100nm). Overlap between z and y is under consideration.

Readout System

Readout system is an electronics which control CCDs, digitize image signal and transfer it to the data acquisition (DAQ) system in control room. Details will be described in the next chapter.

2.3.3 Comparison with other projects

Several wide-field imaging projects for probing dark energy are in progress, which is listed in table 2.2.

The survey speed at which objects can be detected scales as the *etendue* of the telescope, which is defined as the product of the size of its primary mirror (A) times the field of view (Ω).

Pan-STARRS is run by The University of Hawaii. Although the aperture of the telescope is small (1.8 m), using 4 cameras whose FOV is large (3 degrees) provides the high ability of survey. One of the cameras already commenced operation.

The Dark Energy Survey (DES) is run by Fermilab involving the National Optical Astronomy Observatory (NOAO), which succeeds Sloan Digital Sky Survey (SDSS). Since it uses a telescope constructed in 1970's, its imaging ability is worse than Subaru. This is a disadvantage for weak lensing survey.

The Large Synoptic Survey Telescope (LSST) is proposed by SLAC, which has 8.4-m mirror and 7-degree camera. Although its $A\Omega$ overwhelms other projects, its first light is planned in 2014.

Among these projects, the HSC has the largest $A\Omega$ and highest imaging quality at the moment of its first light. The HSC will provide data enough to put new restrictions on the nature of dark energy.

Project	Telescope	$A\Omega$	First Light
HSC	Subaru	91	2011
DES	CTIO	37	2010
Pan-STARRS	New Telescope	13.4 x 4	partly started
LSST	New Telescope	329	2014

Table 2.2: Ongoing Projects for Probing Dark Energy

Chapter 3

Overview of the Readout System for the Hyper Suprime-Cam

The readout system of HSC handles ~ 440 outputs since it employs ~ 110 CCDs and each CCD has 4 outputs. Each CCD has 2048×4096 pixels and the readout electronics digitize the image signal into 16 bits. Thus total data size per exposure becomes ~ 2 Gbytes. It is required to read the entire array in 10 seconds in fast readout mode which corresponds to a data rate of 1.6 Gbits/s. Handling this large amount data at high speed is challenging. The noise of the readout electronics should be smaller than that of a CCD ($\sim 4e^-$)

Number of CCD	~ 110
Total Signal Outputs	~ 440
Data Resolution	16 bits
Image Size	~ 2 Gbytes
Readout Time	min. 10sec
Data Rate	~ 1.6 Gbps
Readout Noise	$\ll 4e^-$

Table 3.1: Electronics Requirements

3.1 Design Concept

Fig.3.1 shows a block diagram of the readout system for the HSC, which consists of four parts: the CCDs, a DAQ system, a Back End Electronics (BEE), a Front End Electronics (FEE)[15].

The DAQ system controls the overall system and gathers image data from the CCDs. The FEE is an analog part, which consists of mainly five parts: pre-amplifiers, correlated double samplings (CDSs) [19], 16-bit ADCs, digital-to-analog converters (DACs), a CCD

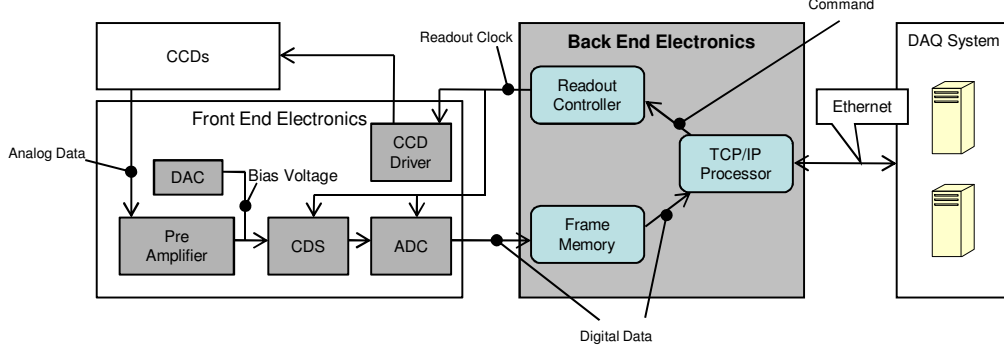


Figure 3.1: A schematic of the readout system for the HSC.

driver. A group consisting of one pre-amplifier, one CDS, one ADC, and one DAC is used for processing a single output from the CCD. The BEE is a digital part which provides the interface between the CCDs, the FEE and the DAQ system. The BEE consists of three blocks: Transmission Control Protocol/Internet Protocol (TCP/IP)[21][22] processor, readout controller, and frame memory. The FEE and the BEE are integrated into the camera unit, while the DAQ system is located in a control room. Since the distance between the FEE and the BEE is large, low voltage differential signalling (LVDS) is used for the signal standard of the communication between the FEE and BEE. Ethernet and TCP/IP are employed for the communication protocol between the BEE and the DAQ system, which has a high degree of reliability and flexibility.

From the DAQ system, commands to control the CCDs and the FEE are sent to the BEE. The readout controller receives these commands through the TCP/IP processor. Following these commands, the readout controller generates readout clocks for the CCDs and the FEE. The readout clock controls the CCDs through the CCD driver which converts the signal level of the readout clock. The analog data is then sent from the CCDs to the FEE. The data is amplified 3-4 fold by the pre-amplifier. The amplified analog data is then sent to the CDS after a bias voltage is added by the DAC, in order to reduce the possibility that the analog image data is cut off from the CDS range when the environment around the CCDs changes. The CDS integrates the data part and the reset part of the analog image signal respectively and outputs the difference of these signals for reducing noise (for details see Fig. 3.2). The integration time of the CDS is controlled by the readout controller of the BEE. The image signal processed by the CDS is then sent to the ADC. The ADC is also controlled by the readout controller of the BEE. After being digitized by the ADC, the image data arrives into the BEE. The whole image data is then stored in the frame memory. The stored image data in the frame memory is transferred to the TCP/IP processor and the image data is transferred to the DAQ system over Ethernet.

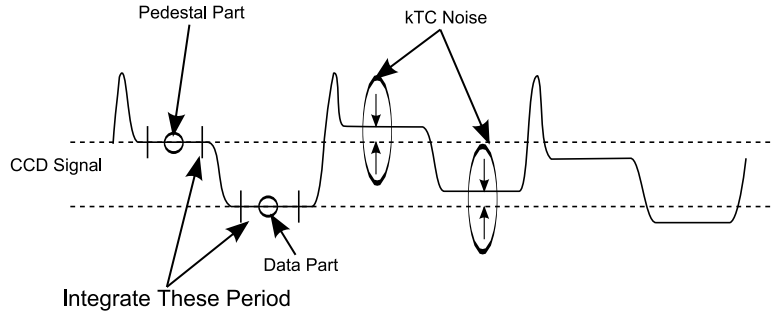


Figure 3.2: Correlated Double Sampling of CCD Signal. In a typical CCD signal, the reset part (corresponding to $0e^-$) comes first, and data part (corresponding to the number of electrons which pixel acquired) next. The issue is that there is noise that adds the bias to the whole signal of one pixel, which is called kTC noise. To cancel this noise, the reset part and the data part are integrated respectively and their difference is output. This process is called correlated double sampling.

3.2 Front End Electronics

The FEE is under development at the NAOJ. It is essentially same as the MFront2 [15] which is already used as the front end electronics of the existing Suprime-Cam. Functionality of the FEE are as follows.

- Power management
 - Voltage, current, and temperature monitor
 - shutdown switch in case of fault
- Bias voltage generation and clock driving
 - Multiple output DAC, CCD driver
- Output signal processing
 - Pre amplifier
 - Double slope type CDS
 - 16bit ADC
- Digital interface
 - CPU
 - LVDS interface for the communication with the BEE

In the current design, one FEE board is used for 4 CCDs. Since the CCD has 4 outputs, the single FEE board handles 16 outputs in parallel. ~ 30 FEE boards are employed for the whole readout system.

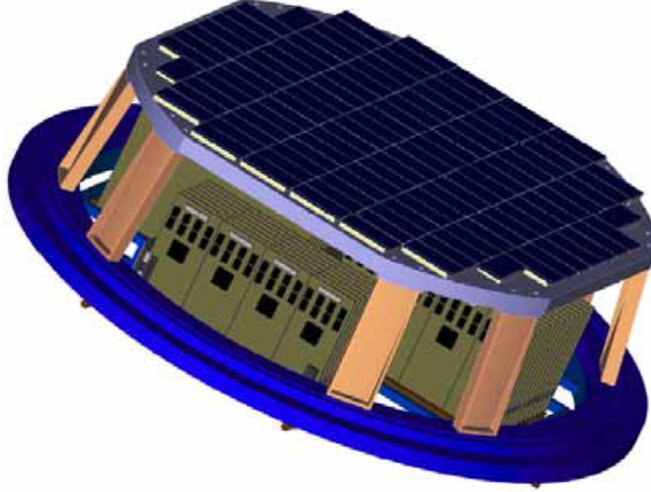


Figure 3.3: FEE Boards Alignment. The FEE boards are located under the cold plate in the cryostat

The FEE is located under the cold plate as is shown in Fig.3.3. Although the cold plane is cooled down to -100 degrees Celsius by pulse-tube coolers[16], the FEE is not actively cooled. It is almost the same temperature of the vacuum cryostat wall. This cooled and vacuumed space is one difference from the FEE of the existing Suprime-Cam. Placing a huge volume of electronics inside vacuumed space and keeping the environment cold is challenging.

3.3 Back End Electronics

The BEE is under development at the University of Tokyo. The BEE is located outside of the cryostat. Fig3.4 shows the schematic diagram of the planned BEE system. The BEE is consists of two Eurocard systems[17][18]. Each Eurocard system consists of the following modules.

- Power Supply Module
This module needs a 5-V power supply and distributes a 5-V and a 3.3-V supply to other BEE modules. The module also pilots the current.
- Readout Module

This module is the main module of the BEE, which has the three functionality of the BEE described in §3.1: the TCP/IP processor, the readout controller and the frame

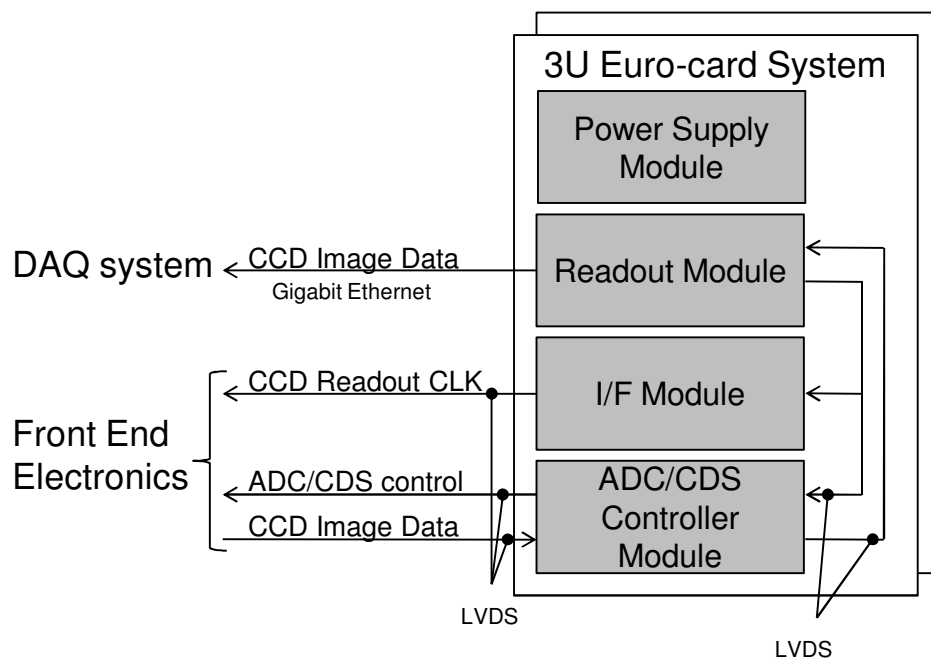


Figure 3.4: Schematics of a planned BEE system

memory. The single readout module controls half of the CCDs and the FEE. The signal standard of the readout clock and the CCD image data is LVDS. One readout module has one Gigabit Ethernet link for the communication with the DAQ system. Since we employ two Eurocard systems, 2 links of Gigabit Ethernet are used.

- I/F Module

This module drives the LVDS CCD readout clock from the readout module again to fair the signals.

- ADC/CDS Controller Module

This module controls the ADCs/CDSs in the FEE with a Serial Peripheral Interface (SPI), and then transfers digitized image data to the readout module through LVDS signal lines.

We have developed a prototype of the readout module, and discuss details of the module in the next chapter.

Chapter 4

R & D of the Readout Module for the Back End Electronics

We have developed the prototype of the readout module which is the main part of the BEE. In this chapter we present the details of the design and architecture of the module, and the result of the performance test.

4.1 Requirement

The requirement for the readout module is as follows:

- **Throughput: 0.8Gbps**

It is required that the readout system should send the whole image data of one exposure (~ 2 Gbytes) to the DAQ system within 10 seconds in the case of the fast readout mode, which corresponds to 1.6 Gbps in throughput. Since we employ two readout modules, the required throughput for the single readout module is 0.8 Gbps.

- **Large capacity of the frame memory ($>\sim 1$ Gbyte)**

It is required that the readout module has a large frame memory capacity in order to store the entire image data (~ 2 Gbytes). Since two readout modules are employed in the whole system, the requirement for the single readout module is the frame memory capacity of more than $>\sim 1$ Gbyte.

- **Noise $\ll 4e^-$ around the bias voltage**

The noise from the readout module must be much lower than the noise of the CCD output which is expected to be $\sim 4 e^-$. However, when the number of electrons is large, the Poisson noise of electron itself is so large that other noise is not important. Thus, the requirement for the noise is that the noise from the readout module should be much lower than that from the CCD in the “few electron region.” The few electron region corresponds to the voltage around the bias voltage which the DAC adds before the CDS in the FEE. To summarize, the requirement of the noise of the readout electronics is $\ll 4e^-$ around the bias voltage.

- **Small, light, and low power consumption**

Since the readout module will be integrated into the camera unit, this module should be small and light, and its power consumption should be low.

4.2 Overview of the readout module

4.2.1 Design

The prototype readout module we have developed is called *GESiCA*: Gigabit Ethernet SiTCP CMC board for Astronomy. This module has the following features: small size, light weight, low power consumption, Gigabit Ethernet, a large memory capacity, and low clock jitter for the readout clocks.

Fig.4.1 and Fig.4.2 are pictures of *GESiCA*. Its size is 149 mm \times 79 mm \times 12 mm and weight is \sim 100 g, which is compatible with the Common Mezzanine Card (CMC) specified as IEEE P1386 [25]. Generally, CMC are designed to be plugged into a slot on a host's mother board. *GESiCA* is integrated into the BEE through a host board in 3U Eurocard size. Note that *GESiCA* is independent of a system of host modules, so various systems can be used in different phases of development. For instance, to evaluate performance, we combined *GESiCA* to a part of the current Suprime-Cam system, whose detail will be described in Chapter 5.

Main parts of *GESiCA* are the CMC connectors, a Gigabit Ethernet Physical Layer Device (PHY; DP83865DVH, National Semiconductor Corp.), a Double-Data-Rate2 Synchronous Dynamic Random Access Memory (DDR2 SDRAM) Small Outline Dual In-line Memory Module (SO-DIMM), a Field Programmable Gate Array (FPGA; XC5LVX110-FFG676-2I, Xilinx Inc.), and a low-jitter clock source.

- **CMC connectors**

GESiCA has three CMC connectors. Two connectors are used for supplying power (5V and 3.3V for each connector).

The other one connector is used for communication signals. All pins of this connector are connected directly to the FPGA. The most pins of this connector are mainly used for the communication between *GESiCA*, the CCDs and the FEE. The readout clock is sent to the CCDs and the FEEs through this connector, and the image data comes back to *GESiCA* also through this connector. The other pins of the connector will be used for the other general purpose such as sending commands to shutter. Pin assignment of this connector is shown in table4.1. Communication protocol adheres to the original specification of the NAOJ. Signal standard is complementary metal oxide semiconductor (CMOS) 3.3V, which are converted to LVDS on the host board in the HSC system.

- **Gigabit Ethernet PHY device**

As an interface between *GESiCA* and the DAQ system, we employ the PHY and one Gigabit Ethernet 1000BASE-T port with Unshielded Twisted Pair cabling (UTP). Since two *GESiCA*s will be employed, two links of Gigabit Ethernet will be used for

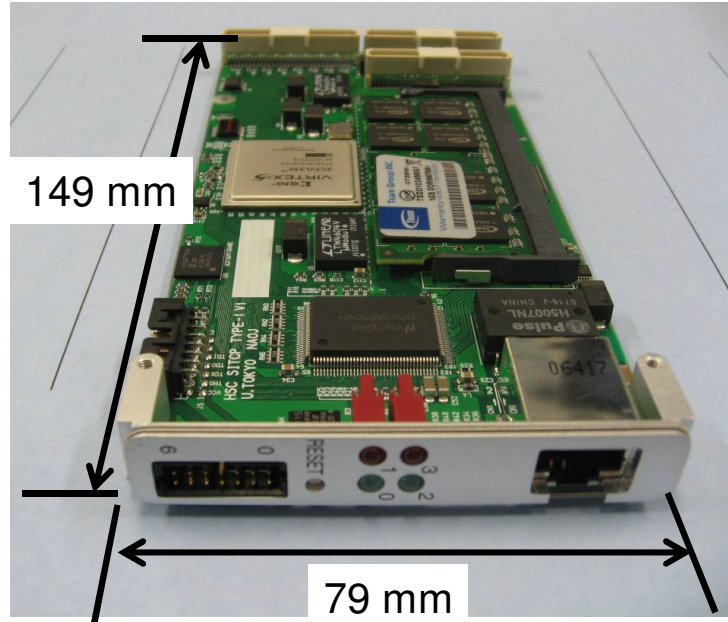


Figure 4.1: Photograph of prototype readout module (GESiCA) (front view)

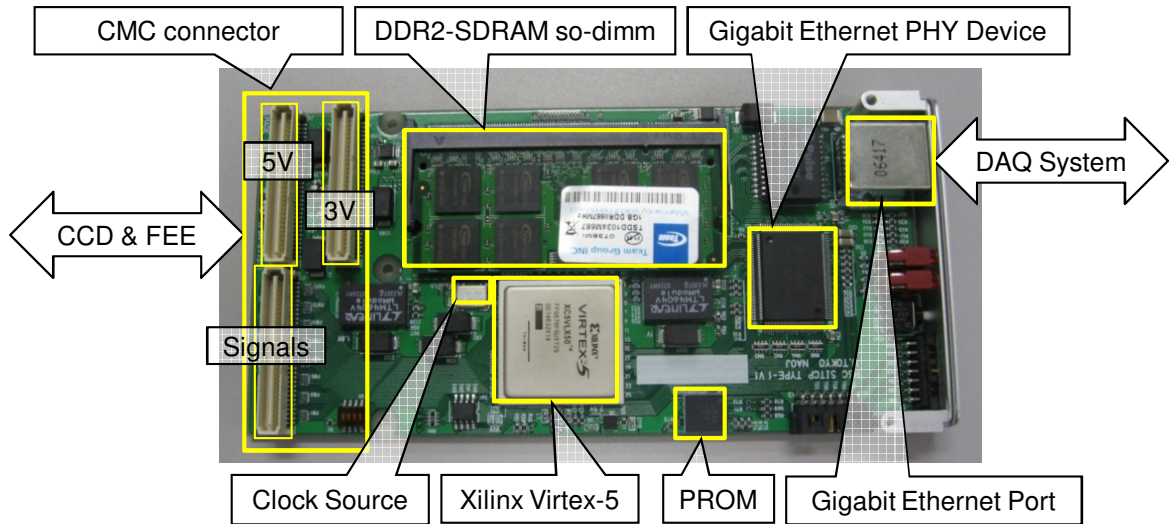


Figure 4.2: Photograph of prototype readout module (GESiCA) (top view)

the HSC readout system, which meets the requirement of throughput for the whole HSC system (~ 1.6 Gbps).

- **DDR2 SDRAM SO-DIMM**

The DDR2 SDRAM SO-DIMM plays the role of the frame memory with the capacity of up to 2 Gbytes. The capacity is enough to store temporarily image data because one GESiCA processes half of the CCDs - the total size is about 2 Gbytes per exposure.

- **FPGA**

The FPGA controls the PHY and the DDR2 SDRAM SO-DIMM, and generates the readout clock for the CCDs and the FEE. The FPGA can be configured by Programmable Read Only Memory (PROM) which can be overwritten by computer through the Joint Test Action Group (JTAG).

- **Low-jitter clock source**

The FPGA generates a 32-bit readout clock to control the CCD and the FEE. We described how the readout clock controls the FEE in §3.1. The jitter of the readout clock which controls the CDS creates noise. In order to make the noise low, it is necessary to use a low-jitter clock source. Thus, we mounted an Epson Toyocom XG-1000CA whose jitter is 3-ps (typical) root mean square on GESiCA. The result of the test for the noise will be described in §4.4.3.

4.2.2 Operation

Fig.4.3 is a block diagram of GESiCA. The FPGA contains three blocks: a hardware-based TCP/IP processor (SiTCP) [26], readout controller, and frame memory controller. The PHY and SiTCP are connected with the standard interface, Gigabit Media Independent Interface (GMII) specified by IEEE 802.3[23]. Receiving commands from the DAQ system through SiTCP, the readout controller generates 32-bit CMOS 3.3V readout clocks which controls CCD and FEE through the CMC connectors. The frame memory controller then grabs 16-bit digitized image data from the FEE through the CMC connectors and writes the image data to the frame memory (DDR2 SDRAM). The image data is then read from the frame memory and sent to DAQ system through SiTCP. It should be noted that the DDR2 SDRAM is controlled in the clock frequency of 50 MHz in GESiCA, while the standard requires 125 MHz or higher[29].

4.2.3 Communication with the DAQ system

GESiCA can be seen as a TCP server from the DAQ system. The user can control GESiCA by using a socket software.

The image data is sent from to the DAQ system over Transmission Control Protocol (TCP), while other communication with the DAQ system, including commands to the the readout controller, are established over User Datagram Protocol (UDP) [20]. TCP has the high reliability of data transfer. UDP sends packets as quick as a user commands. Thus

Pin Number	Signal	Pin Number	Signal
1	P_IN0	33	CP0
2	RESET	34	CP1
3	P_IN1	35	CP2
4	SYNC0	36	CP3
5	P_IN2	37	CP4
6	ACTIVE	38	CP5
7	P_IN3	39	CP6
8	SYNC1	40	CP7
9	P_OUT0	41	CP8
10	UART0	42	CP9
11	P_OUT1	43	CP10
12	FG_SCLK	44	CP11
13	P_OUT2	45	CP12
14	UART1	46	CP13
15	P_OUT3	47	CP14
16	FG_LOAD	48	CP15
17	FG0	49	CP16
18	FG1	50	CP17
19	FG2	51	CP18
20	FG3	52	CP19
21	FG4	53	CP20
22	FG5	54	CP21
23	FG6	55	CP22
24	FG7	56	CP23
25	FG8	57	CP24
26	FG9	58	CP25
27	FG10	59	CP26
28	FG11	60	CP27
29	FG12	61	CP28
30	FG13	62	CP29
31	FG14	63	CP30
32	FG15	54	CP31

Table 4.1: Pin assignment of the CMC connector of GESiCA. PIN0-3/POUT0-3 are general purpose inputs/outputs. RESET is used for master reset. ACTIVE is used for return from sleep. SYNC0-1 are used for synchronization of readout modules. UART0-1 are used for serial communication. The signals mentioned above will be used in the future. FG_SCLK and FG_LOAD are the control signal from the FEE. FG0-15 is the 16-bit digital image data from the FEE. CP0-31 is the 32-bit readout clock for the CCDs and the FEE.

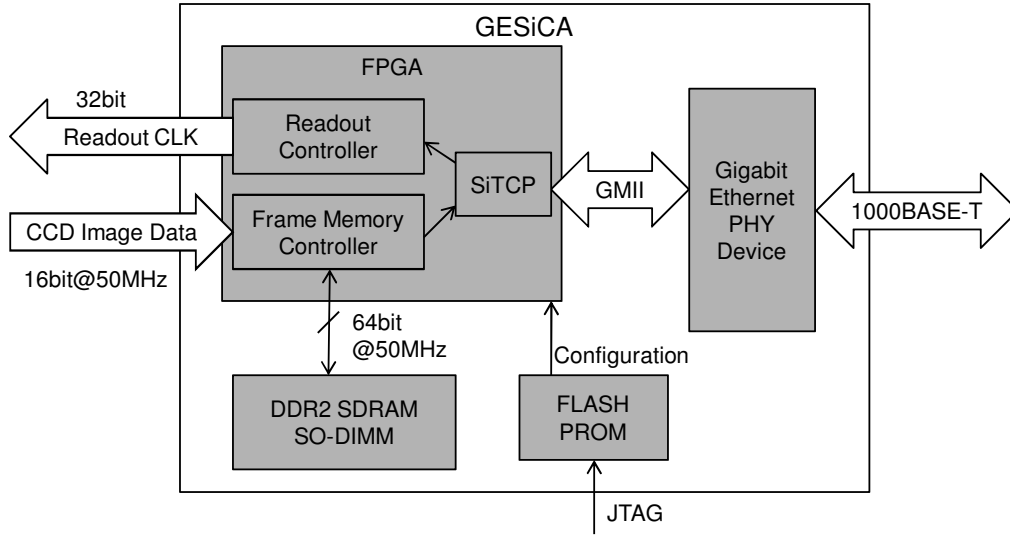


Figure 4.3: A block diagram of GESiCA

safety of image data transfer is insured and commands can be sent on the timing a user indicates.

Communication over TCP

While TCP connection is established, the DAQ system can receive the image data from GESiCA. The header is added to the beginning of each exposure. The specification of the header is shown in Fig.4.4. Version indicates the current GESiCA version. Data Length indicates the number of the whole image data in units of byte. Frame ID is incremented everytime new exposure is taken. R0-R6 are the registers a user can configure freely, which is prepared for future use. All parameters are in network byte order (big endian).

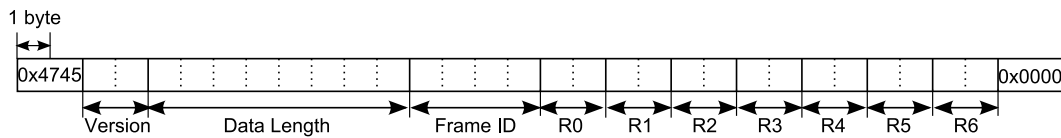


Figure 4.4: Image Data Header Specification. The size of the header is 32 bytes.

Communication over UDP

Using communication over UDP, a user can edit the configuration of GESiCA and send commands to GESiCA.

- configuration

- exposure size
It is necessary to set the the number of pixels per exposure in order to tell GESiCA where it should add the header of the image data.
- readout clock pattern
GESiCA stores patterns of the readout clock inside. A user can configure the patterns of the readout clock.
- command
 - invoke
Start a pattern of the readout clock.
 - reset
Stop the readout clock. All outputs goes to low level.
 - busy?
This command is used to ask the state of GESiCA. If the readout controller generates the readout clock, GESiCA returns busy. If not, it returns idle.

Details of these configuration and commands are described in Appendix.A.2.

4.3 FPGA implementation

In the following sections, we will explain details about the FPGA implementation.

4.3.1 Clock Management

Fig.4.5 shows clock distribution in the FPGA. The clock distributor which contains four Digital Clock Managers (DCMs) built in the FPGA[28] provides all the clocks used in the FPGA. Referencing the input clock (100MHz; Epson Toyocom XG-1000CA), a DCM generates a clock whose period and phase are specified by a user. The clock distributor provides the following clocks.

- **SYS_CLK**
The frequency of SYS_CLK is 130MHz. This clock is used as a system clock of the FPGA which is distributed to the SiTCP, readout controller, and frame memory controller.
- **GMII_CLK**
The frequency of GMII_CLK is 125MHz. This clock is provided to SiTCP and used as the GMII clock for communication with the PHY device.
- **DRAM_CLK0, DRAM_CLK90**
The frequency of these clocks is 50MHz. The phase of DRAM_CLK90 is shifted by 90 degrees compared to DRAM_CLK0. These clocks are provided to the frame memory controller and used to control the DDR2 SDRAM SO-DIMM.

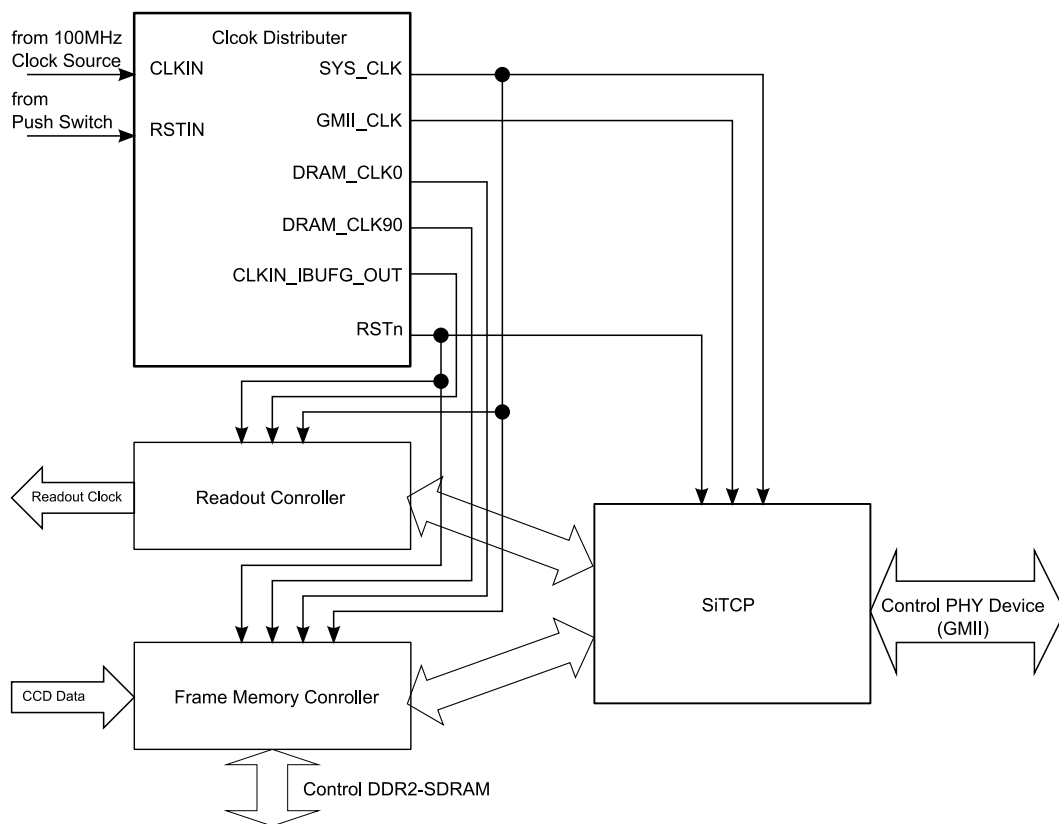


Figure 4.5: Clock Distribution in FPGA

- **CLKIN_IBUFG_OUT**

CLKIN_IBUFG_OUT outputs the clock from the clock source without any modification. This clock is provided to the readout controller and used for the readout clock to make the most of the low-jitter clock source and minimize the noise of the electronics (see §4.2.1).

- **RSTn**

RSTn is an active-low reset signal distributed to the SiTCP, readout controller and frame memory controller. While RSTIN, which is connected to an active-low push switch on GESiCA, is negated, all of the DCMs are reset and the RSTn is negated, that is, the reset signal propagates over the FPGA. After the RSTIN is asserted and initializations of all the DCMs are completed, the RSTn is asserted.

4.3.2 SiTCP

SiTCP is a hardware-based TCP/IP processor[26], which provides the Gigabit Ethernet connection between the DAQ system and GESiCA. SiTCP offers a solution for the following requirements:

- Throughput: 0.8Gbps
- Small, light, and low power consumption

Generally, TCP/IP is processed by software running on a standard operating system. A powerful hardware and CPU are required to process these protocols at gigabit rates, and thus power consumption becomes of order several tens of watts and the size and weight are both large. SiTCP is able to overcome these difficulties. The logic size of SiTCP is small enough to allow us to implement it on a single FPGA. This feature enables us to design a small, light and low-power consumption module that has high speed data transfer capability at the same time.

Fig.4.6 represents I/O of SiTCP. SiTCP controls the PHY Device through GMII and has user interface signals for TCP and UDP whose details are shown in table 4.2. To communicate with GESiCA over UDP, a user should use an original protocol called Remote Bus Control Protocol (RBCP), which is encapsulated in UDP packets. Details of RBCP are described in Appendix.A.1.

4.3.3 Readout Controller

Functionality

The readout controller generates a 32-bit readout clock to control the CCDs and the FEE, following commands from the DAQ system. This module can also generate additional a 4-bit control clock for general purpose (P_OUT[3:0] in Appendix.A) which will be used in the future. Hereafter, the 32-bit readout clock + 4-bit general purpose clock is simply called the readout clock. These signals are sent to the outside of GESiCA through the a CMC connector.

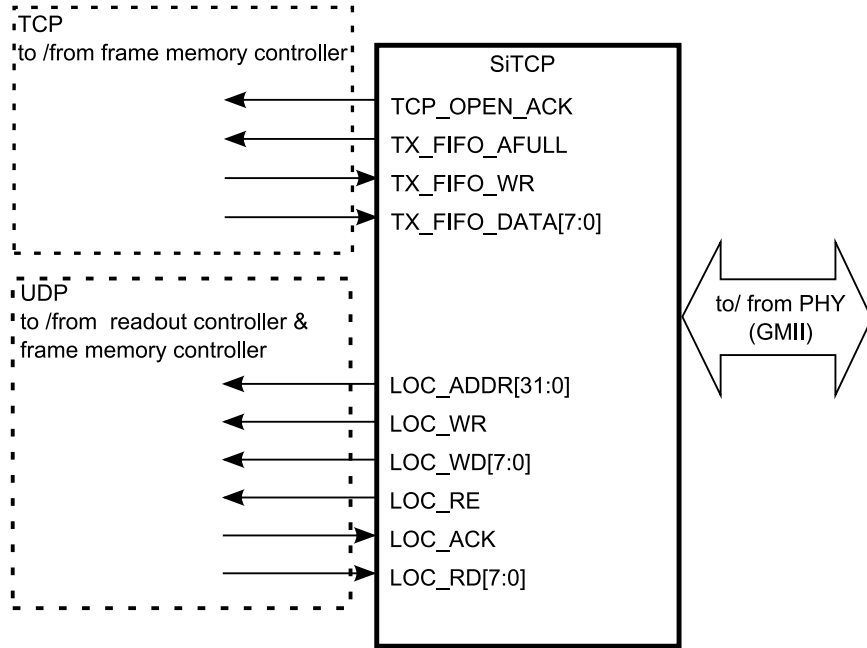


Figure 4.6: I/O of SiTCP

Category	Signal	Description
TCP	TCP_OPEN_ACK	Acknowledge: When asserted, TCP connection is established.
	TX_FIFO_AFULL	Almost full flag: When asserted, the buffer for TX_FIFO_DATA in SiTCP can perform only 8 more write.
	TX_FIFO_WR	Write enable: If the buffer in the SiTCP is not full, asserting this signal causes data (on TX_FIFO_DATA) to be written to the buffer
	TX_FIFO_DATA	8-bit width write data
UDP	LOC_ADDR	32-bit width address
	LOC_WR	Write enable
	LOC_WD	8-bit width write data
	LOC_RE	Read enable
	LOC_ACK	Access acknowledge from target devices. In read access, valid read data
	LOC_RD	8-bit width read data

Table 4.2: User Interface Signals of SiTCP

The readout controller provides the following solution for the requirement presented in §4.1.

- low-jitter readout clock

As mentioned in §3.1 and §4.2.1, the low-jitter readout clock for controlling integration period of the CDS is crucial to realize low noise. The readout controller uses the clock directly from the low-jitter clock source to generate low-jitter readout clocks (cf. §4.2.1). The estimation of the noise resulting from the jitter will be described in §4.4.3.

In addition, the readout controller has the following features

- fine tuning capability of the readout clock

The readout clock is generated in units of 10ns, which is much finer than that of the existing Suprime-Cam system (80 ns). It is required to make the readout clock of the HSC about two times faster than that of the Suprime-Cam. Empirically, it is known that faster readout speed induces the larger noise. The fine tuning capability of the readout clock allows us to optimize both the readout clock speed and the noise.

- programmable readout clock

The readout controller has large RAM for storing the readout clock. Following the commands from the DAQ system, the readout clock is read from the RAM and then sent to the CCDs and the FEE. The readout clock can be configured by writing new readout clock patterns on the RAM from the DAQ system.

Clock Pattern Description File

To configure the readout clock, clock pattern description (CPD) files are used. A CPD file is compiled in the DAQ system and sent to the readout controller in GESiCA as UDP packets (for details see AppendixA). We will explain how a CPD file is stored in the RAM in the readout controller in the next section.

CPD files are written in the same format as the Modularized Extensible System for Image Acquisition (MESSIA) series [27] which is already employed in a number of optical and near-infrared instruments throughout the Japanese astronomical community, including Suprime-cam. This compatibility makes it easy for many users to program the clock patterns since the format is widely used.

Here is an example of CPD files. A CPD file can be divided into two parts: procedure definition (PD) part and clock pattern definition (CPD) part. The PD part defines the clock procedure which arranges clock patterns defined in the CPD part. We explain keywords for these two parts.

Example of CPD File

```
###PD Part###
begin proc1                # procedure definition: "proc1"
    ccd_operation 0 1 100  # signal output: iterate ope "1" 100 times.
```

```

        ccd_operation 0 2 10    # signal output: iterate ope "2" 10 times.
    end

begin proc2
    loop3_start 42_0000_0000
        ccd_operation 0 1 5
        ccd_operation 0 2 3
    loop3_continue
end

###CPD Part###
set_default_bit      1          # omitted bits are regarded as 1
set_tick_prec        20         # tick precision is 20ns

set_clock_tick       3          # clock tick is 3 * 20ns (set_tick_prec'ed)
operation_type        1         # signal definition: ope "1"
start                 35 32 0 5 4 10
t   4                 | | ] ] | |
t   1                 | ] | ] ] |
end

set_clock_tick        1
operation_type         2
start                 35 32 0 5 4 10
t   2                 ] | ] ] ] |
end

```

- Keywords for the PD part
Keywords for the PD part is called *opcode*. The readout clock defined from 'begin' to 'end' is called *procedure*.
 - begin [label]
This word indicates the beginning of a *procedure* with the name of [label].
 - ccd_operation 0 [ope_type] [iteration]
This word indicates generation of the clock pattern defined as operation_type [id] in the CPD part with the clock pattern repeating [iteration] times. [iteration] runs from 1 to $2^{32}-1$. [ope_type] corresponds to [id] of the 'operation_type'.
 - loop1_begin [count] / loop2_begin [count] / loop3_begin [count]
This word indicates the beginning of a loop which repeats [count] times. [count] runs from 1 to $2^{32}-1$. A user can set up to triple nested loops.
 - loop1_continue / loop2_continue / loop3_continue
This word indicates the jumping-off point to "loop[1-3]_begin [count]".

- end
This word indicates the end of the *procedure*.
- nop [duration]
This word indicates the no operation for $10 \times [\text{duration}]$ ns. [duration] runs from 1 to $2^{32}-1$.
- Keywords for the CPD part
 - set_default_bit [siglevel]
This word sets the signals omitted in 'start [bit0] ([bit1] (...))' below to low ('0') or high ('1') level, following [siglevel] = 0 or 1, respectively.
 - set_tick_prec [nsec]
This word sets the minimum time units used in all clock patterns. [nsec] must be multiples of 10.
 - set_clock_tick [length]
This word sets the time unit used in a clock pattern defined by 'operation_type [id]' just after this. The time unit is set to $[\text{length}] \times [\text{nsec}]$ ns ([nsec] is set by 'set_tick_prec [nsec]').
 - operation_type [id]
This word defines a clock pattern which has a name of [id].
 - start [bit0] ([bit1] (...))
This word indicates the start of the clock pattern. This word also declares the bits of the readout clock that a user want to use by [bit0] ([bit1] (...)). There are bits 0-35 and the order of the bits is not cared. Unused signals can be omitted and fixed to the signal level set by 'set_default_bit [siglevel]' described above.
 - t [duration] [val0] ([val1] (...))
This word describes a logic level of each bit of the readout clock declared in 'start [bit0] ([bit1] (...))'. 'l' means low level and 'h' means high level. [duration] means sustained period of the logic levels in units of $[\text{length}] \times [\text{nsec}]$ ns. 't [duration] [val0] ([val1] (...))' is written in time series from top down. $[\text{duration}] \times [\text{length}] \times [\text{nsec}]$ should be from 1 to $2^{18}-1$.
 - end
This word indicates the end of the clock pattern.

Control from the DAQ System

The readout controller is controlled by receiving commands of 'invoke', 'reset', and 'busy?' from the DAQ system (cf. §4.2.3). 'Invoke' starts a *procedure* described in a CPD file. Software to send the commands to GESiCA is necessary.

The readout clock can be configured by the DAQ system. Software which compiles a CPD file and sends it to GESiCA is developed, though we do not explain its detail here.

The communication protocol between the DAQ system and GESiCA is described in AppendixA.

Architecture

Fig.4.7 is an architecture of the readout controller which consists of a packet interpreter, a pattern generator, and a pattern sequencer. The readout controller is operated in two clock regions; SYS_CLK (130 MHz) and CLKIN_IBUFG_OUT (100MHz). The latter directly comes from the clock source on GESiCA in order to realize low-jitter readout clock.

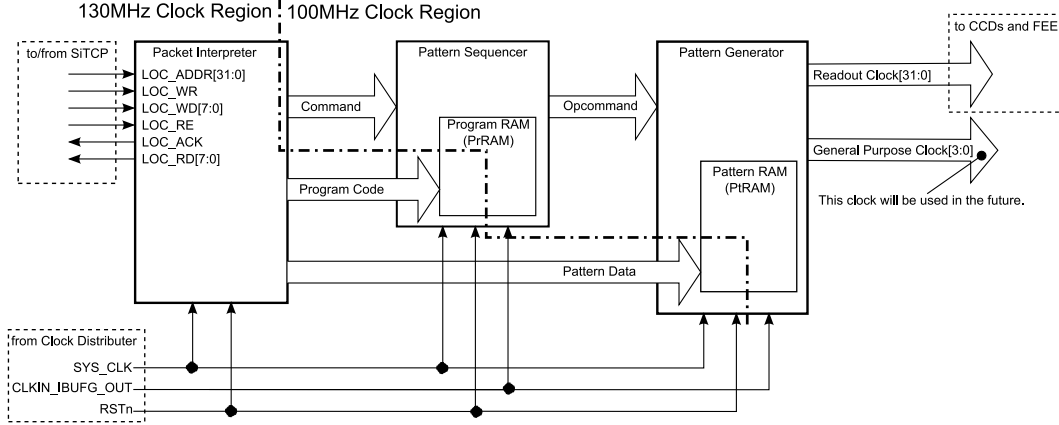


Figure 4.7: Architecture of the Readout Controller.

A packet interpreter translates commands ('invoke', 'stop', and 'busy?') and a readout clock configuration which come from the DAQ system as UDP packets. The commands are sent to the pattern sequencer. At this time, the commands is entered into the 100-MHz clock region from the 130-MHz clock region, flipped by three flip flops whose clock frequency is 100 MHz. The readout clock configuration is divided into two parts in the packet interpreter: one corresponding to a PD part in a CPD file and the other corresponding to a CPD part. The former is sent to a program RAM (PrRAM) in the pattern generator as a program code, while the latter is sent to a pattern RAM (PtRAM) in the pattern generator as pattern data. Writes made into these RAMs are operated at the clock frequency of 130 MHz (SYS_CLK).

The pattern generator contains the PtRAM for storing clock patterns written in a the CPD part of a CPD file. The size of the PtRAM is 54-bit \times 2048-word. The PtRAM consists of dual-port block RAMs in the FPGA [28]. The dual-port RAM divides a clock region: writing the RAM is operated at the clock frequency of 130 MHz, while reading is in 100 MHz. The structure of a word of the PtRAM is described in table 4.3. Each bit of [Bit Pattern] corresponds to each bit defined in 't [duration] [val0] ([val1] (...))' (signal levels of unused bits are fixed by 'set_default_bit [siglevel]') in the CPD part. [Persistence Time] is a sustained period of [Bit Pattern] for [duration] \times [length] \times [nsec] (they are defined in the CPD part) ns.

The pattern sequencer contains the PrRAM to store *procedures* of the readout clock written in the PD part of a CPD file. The size of the PrRAM is 72-bit \times 2048-word. The PrRAM also consists of dual-port block RAMs to divide clock region into 130 MHz and 100 MHz in the same way as the PtRAM. A word in the PrRAM is called *opword* whose

bits	53:18	17:0
	Bit Pattern[35:0]	Persistence Time[17:0]

Table 4.3: Structure of the Memory of the Pattern Generator

structure is shown in table 4.4.

	verb	arg1				arg2			
bits	71:64	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
ccdop	0x03	address		length		iteration			
lstart	0x01	ID	0x000000			the number of loops			
lcont	0x02	ID1	ID2	ID3	0x00	0x00000000			
end	0x00	0x0000000000000000							
nop	0x00	duration				0x00000000			

Table 4.4: Structure of the Memory in the Pattern Sequencer

- **ccdop**
'ccdop' corresponds to 'ccd_operation 0' of *opcode*. [address] indicates the start address of the clock pattern on the PtRAM in the pattern generator, and [length] is the length of a clock pattern which is equal to the number of lines of 't [duration] [val0] ([val1] (...))' in a CPD part of a CPD File. Thus [address] + [length] - 1 corresponds to the end address of the clock pattern. [iteration] of this word corresponds to the [iteration] of *opcode* 'ccd_operation 0'.
- **lstart**
'lstart' corresponds to 'loop[x].start' of *opcode*, where [x]=1, 2, 3. [ID] takes the value of 0x01, 0x02, 0x03, and each value corresponds to the [x] of 'loop[x].start', respectively. [the number of loops] of this word corresponds to the [count] of 'loop[x].start' of *opcode*.
- **lcont**
'lcont' corresponds to 'loop[x].continue' of *opcode*, where [x]=1, 2, 3. When 'loop[x].continue' of the *opcode* emerges continuously in a PD part, they are merged into the single *opword* 'lcont'. [ID[y]] of 'lcont', where [y]=1, 2, 3, takes the value of 0x01, 0x02, 0x03. [ID1] should be the number of the [x] of the innermost 'loop[x].continue', while [ID3] should be the [x] of the outermost 'loop[x].continue'. When the number of continuous 'loop[x].continue' is less than 3, unused [ID[y]] must be 0x00.
- **end**
'end' corresponds to 'end' of *opcode* which indicates the end of a *procedure*.

- **nop**
'nop' corresponds to 'nop' of *opcode*. The pattern sequencer stops for $10 \times [\text{duration}]$ ns.

Let us describe how a command from the DAQ system is processed and readout clock is generated by the readout controller. Here Fig.4.7 will help your understanding. The 'invoke' command from the DAQ system indicates the address of the PrRAM corresponding to the top of a *procedure*. Following this address, the pattern sequencer reads *opwords* from the PrRAM and interprets them into *opcommands*. Then the pattern sequencer sends the *opcommands* to the pattern generator until a *opcode* of 'end' emerges. Following the *opcommands* from the pattern sequencer, the pattern generator reads clock patterns from the PtRAM, and finally readout clocks are generated. The pattern generator has a buffer after the input of *opcommand* and a FIFO before the output of readout clock to realize the gap-less readout clock during a *procedure*.

4.3.4 Frame Memory Controller

Functionality

The frame memory controller grabs the image data from the FEE and writes the data into the frame memory (DDR2 SDRAM). Then the frame memory controller reads back the data from the frame memory and sends it to the SiTCP. The frame memory is regarded as a large buffer.

The frame memory controller + the frame memory has the following functionalities which solve the requirements presented in §4.1.

- **Throughput $> 0.8\text{Gbps}$**
Although Gigabit Ethernet is employed for the communication between GESiCA and the DAQ system, to realize the throughput $> 0.8\text{ Gbps}$, it is necessary for the frame memory controller to mask the access to the DDR2 SDRAM from outside as if the input from the FEE is directly connected to the DAQ system. The frame memory controller controls DDR2 SDRAM with arbitrating the image data from the FEE and that to SiTCP in order to realize this mask.
- **Large memory capacity ($> \sim 1\text{ Gbyte}$)**
The frame memory controller can control the DDR2-DRAM SO-DIMM with the capacity up to 2 Gbytes.

In addition to these solutions, the frame memory controller has the functionality of adding the header to the exposure, referring to the parameter "exposure size" which a user can configure from the DAQ system (cf. §4.2.3).

I/O of the frame memory controller

Fig.4.8 shows the I/O of the frame memory controller. The I/O is classified into five groups

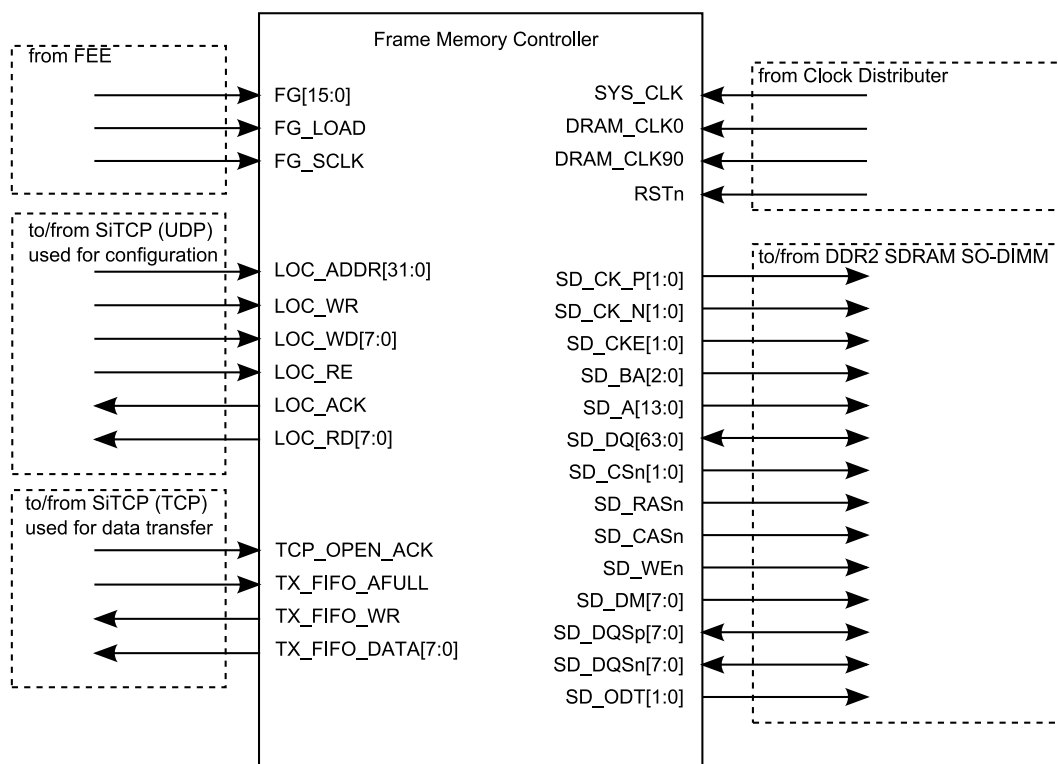


Figure 4.8: I/O of the Frame Memory Controller

- from FEE

When the 16-bit image data (FG[15:0]) arrives, FG_LOAD rises and FG_SCLK is asserted as shown in Fig.4.9. The image data can be grabbed by using the rising edge of the FG_LOAD.

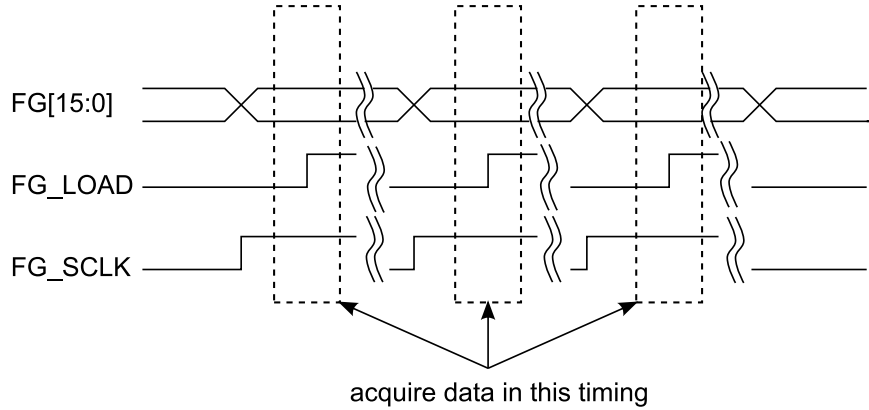


Figure 4.9: Timing Chart of the Signal from the FEE

- to/from SiTCP (UDP)

This group is used for receiving the parameter “exposure size” (cf. §4.2.3) in the form of RBCP packets. For details of communication protocol between GESiCA and DAQ system, see Appendix.A.

- to/from SiTCP (TCP)

This group is used for transferring the image data to the SiTCP. TX_FIFO_DATA[7:0] is used for sending the image data from the frame memory controller to the SiTCP. TX_FIFO_WR is the valid signal for TX_FIFO_DATA[7:0]. TCP_OPEN_ACK is asserted while TCP connection between GESiCA and the DAQ system is established. TX_FIFO_AFULL is asserted when a buffer in the SiTCP which stores TX_FIFO_DATA[7:0] can perform only 8 more write.

- from Clock Distributer

130-MHz SYS_CLK is a clock used for the system clock of the frame memory controller. DRAM_CLK0 and DRAM_CLK90 are used for controlling the DDR2_SDRAM via the frame memory controller. RSTn is active-low reset signal.

- to/from the DDR2 SDRAM SO-DIMM

This group is used for controlling the DDR2 SDRAM SO-DIMM. For specification of the signals, see the DDR2 SDRAM SO-DIMM specification [30].

Basic Architecture of the Frame Memory Controller

Fig. 4.10 shows the architecture of the frame memory controller which consists of the following modules.

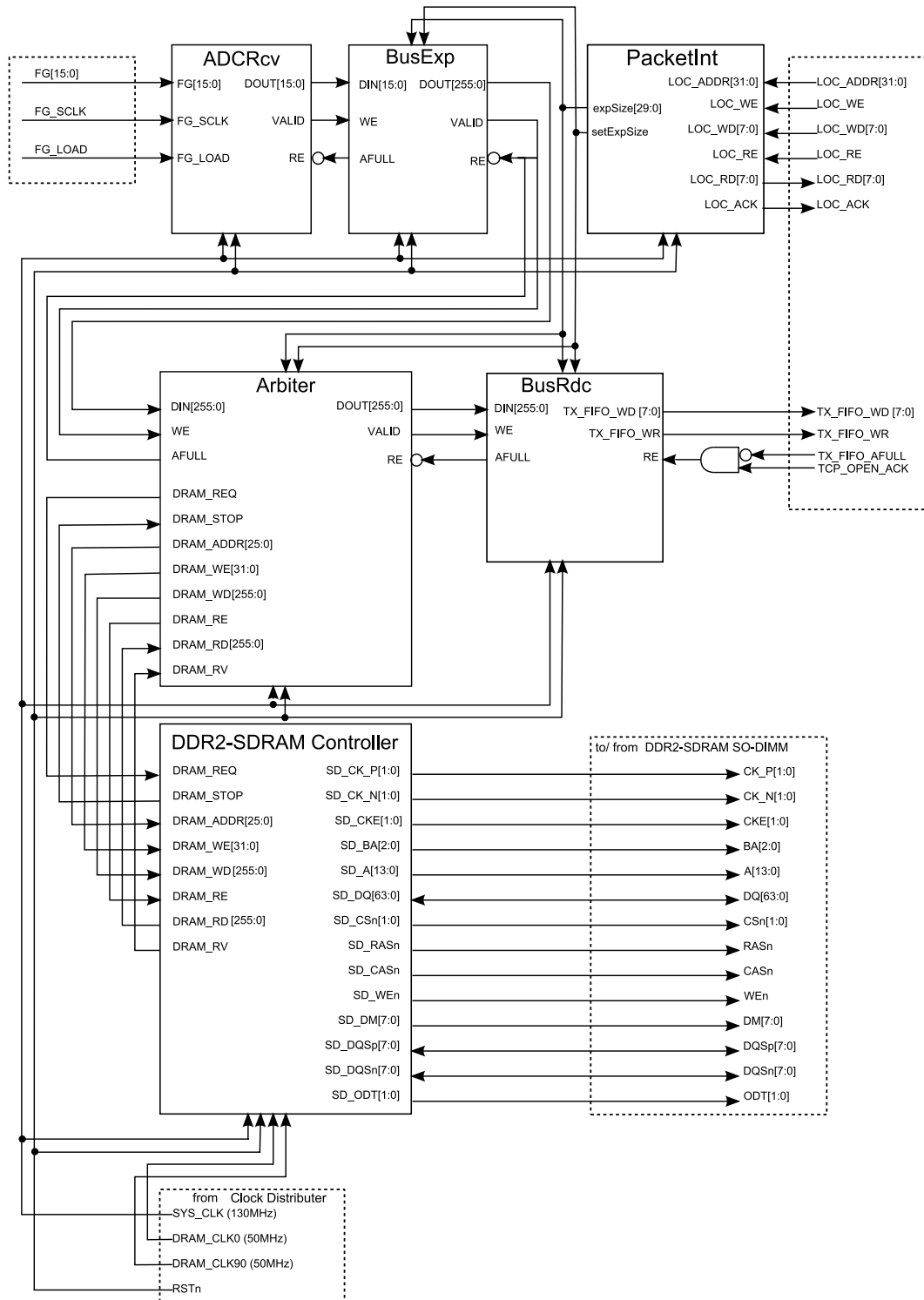


Figure 4.10: Schematic of Frame Memory Controller

- **PacketInt**
This module interprets the RBCP packets from the SiTCP into the parameter “exposure size” and set it to BusExp, Arbiter, and BusRdc.
- **ADCRcv**
This module receives image data (FG[15:0]), using the rising edge of FG_LOAD and FG_SCLK (see Fig.4.9), and then synchronizes the data to SYS_CLK. The data is sent to the BusExp from DOUT[15:0] with asserting VALID, when read enable (RE) is high level.
- **BusExp**
This module combines the 16-bit data from ADCRcv (DIN[15:0]) into 256-bit data (DOUT[255:0]). This module receives the data (DIN[15:0]) when write enable (WE) is asserted. Asymmetric built-in FIFOs of the FPGA[28] are used to expand the bus size from 16 bits to 256 bits. This FIFO receives 16-bit data. When 16 words of the 16-bit data is stored, it outputs 256-bit data as 1 word. If only one more write can be performed before the FIFO is full, almost full (AFULL) is asserted. The inverse of AFULL is connected to the RE of ADCRcv. The 256-bit data is sent to the arbiter with asserting VALID when RE of BusExp is high level. If the number of bits per exposure is not multiples of 256, this module adds dummy data to the end of exposure before the FIFO and fits the size of data to multiples of 256bits, referring to “exposure size” set by the PacketInt.
- **Arbiter**
This module is the key to realize the requirement for the throughput (0.8Gbps). It controls the DDR2 SDRAM controller with arbitrating the demand from the BusExp (writing the data into the frame memory) and the BusRdc (reading the data from the frame memory). Details of this module will be described later.
- **DDR2 SDRAM Controller**
This module controls the frame memory. After receiving the read/write command, address, and data from the arbiter, this module access the frame memory automatically. Thanks to this module, complicated access to the DDR2 SDRAM is masked. Note that this module controls the DDR2 SDRAM at a clock frequency of 50MHz, although the DDR2 SDRAM generally runs at a clock frequency of 125MHz or higher. This feature reduces power consumption. Architecture of this module will be described later.
- **BusRdc**
This module divides the 256-bit data from the arbiter into the size of TX_FIFO_WD (8bits). This module receives the data (DIN[255:0]) when WE is asserted. Asymmetric built-in FIFO is also used in this module for reducing the bus width from 256 bits to 8 bits. This module has AFULL which is asserted when only one more write can be performed before the FIFO is full. The dummy data added by the BusExp is discarded, referring th “exposure size” set by the PacketInt. BusRdc sends

the 8-bit data as TX_FIFO_WD[7:0] to SiTCP with asserting TX_FIFO_WR when TX_FIFO_AFULL is low level and TCP_OPEN_ACK is high.

DDR2 SDRAM Controller

For details of the DDR2 SDRAM, see the specification document[29]. In this section *italic letters* represent the commands sent to the DDR2 SDRAM defined in the specification document.

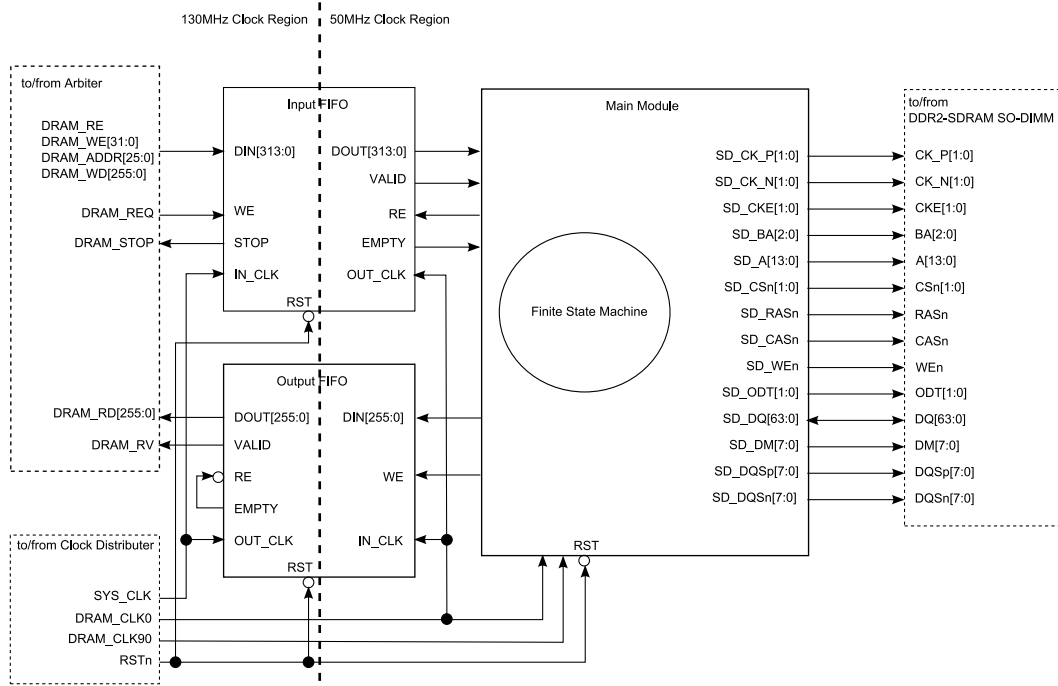


Figure 4.11: Architecture of the DDR2 SDRAM controller

The architecture of the DDR2 SDRAM controller is shown in Fig. 4.11. The DDR2 SDRAM controller consists of 3 modules: input FIFO, output FIFO, and main module. The input FIFO and output FIFO divide clock regions: SYS_CLK is used for the outside of the DDR2 SDRAM controller and DRAM_CLK0 is for the inside. The input FIFO receives the command (DRAM_RE/DRAM_WE[31:0]), address (DRAM_ADDR[25:0]), and data (DRAM_WD[255:0]) as DIN[313:0]. DRAM_REQ is connected to the write enable (WE) of the FIFO. To write data, the address and data should be input with asserting DRAM_REQ and arbitrary bits of DRAM_WE[31:0]. Each bit of DRAM_WE[31:0] corresponds to the data mask (DM[7:0]) of the DDR2 SDRAM, that is, asserting DRAM_WE[x] is the write enable of DRAM_WD[8x+7:8x], where $0 \leq x \leq 31$. To read data, the address should be input with asserting DRAM_REQ and DRAM_RE. The data is read from the DDR2 SDRAM via the main module and stored in the output FIFO, and then the data (DRAM_RD[255:0]) is sent from the FIFO with asserting valid signal (DRAM_RV). The depth of the input

FIFO and output FIFO is 32. If only 4 more write can be performed before the input FIFO is full, DRAM_STOP is asserted.

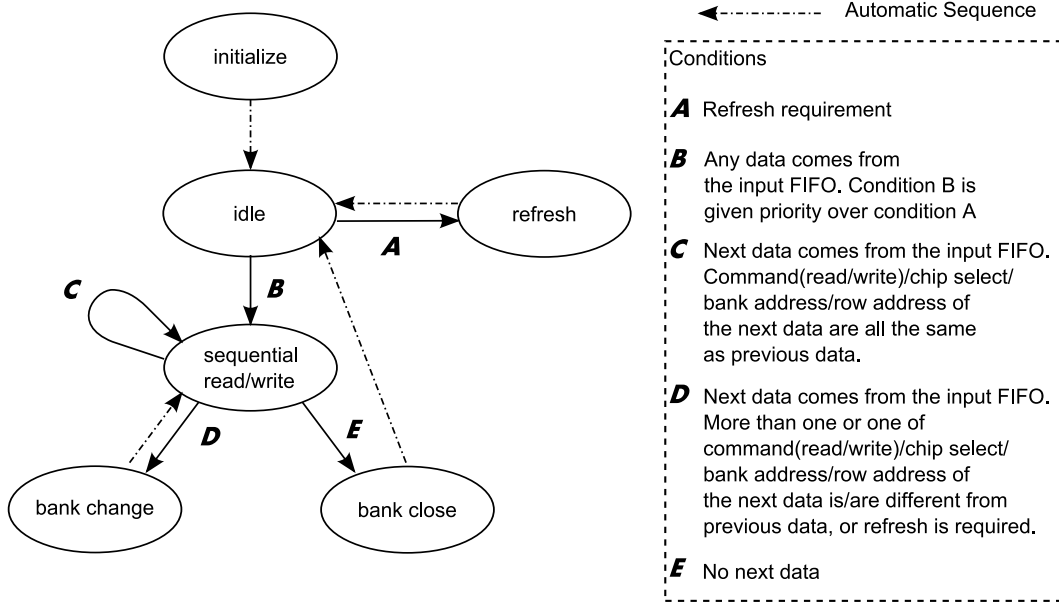


Figure 4.12: State Transition Diagram of the DDR2 SDRAM controller

Inside the main module, there is a finite state machine whose state transition diagram is shown in Fig.4.12. After power up, the main module is in the state of “initialize” and initializes the DDR2 SDRAM. The initialization settings are shown in table 4.5. After initialization, the state moves to “idle” and waits data from the input FIFO. There is a timer for *refresh* in the main module. When the timer requires *refresh* during “idle”, the state changes to “refresh” and *precharge* and *refresh* are sent to the DDR2 SDRAM. After that, the state returns to “idle”. If any data comes from the input FIFO during “idle”, the main module sends *bank active* and its state changes to “sequential read/write” with sending *read* or *write*. If there is a next data from the input FIFO, the state remains same and the main module keeps on sending *read* or *write* to the DDR2 SDRAM. However, if more than one or one of command (DRAM_RE, DRAM_WD[31:0]), chip select, bank address, row address of the next data is/are different from the previous data, or *refresh* is required, the state changes to “bank change” and *precharge*, *refresh*, and *bank active* are sent in this order so as to change banks. After that the state returns to “sequential read/write” with sending *read* or *write* to the DDR2 SDRAM. If there is no data from the input FIFO during “sequential read/write”, the state changes to “bank close” and *precharge* and *refresh* are sent to the DDR2 SDRAM. After that, the state changes to “idle” and the main module waits a new data from the input FIFO.

Fig.4.13 shows a state transition of “bank change” → “sequential read/write” → “bank change”. Here we define this sequence as “bank open”. After the DDR2 SDRAM controller sends the single *read/write* to the DDR2 SDRAM, 256-bit data is divided into 4 bursts of 64-bit data (DQ[63:0]) and sent from/to the DDR2 SDRAM. In this chart, the DDR2

Parameter	Description
DLL	Enable
Active Power Down Exit Time	Fast Exit
Write Recovery for Autoprecharge	2
Mode	Normal
CAS Latency	3
Burst Type	Sequential
Burst Length	4
Qoff	Output Buffer Enabled
RDQS Enable	Disable
/DQS	Enable
Rtt	50Ω
Additive Latency	0
Output Driver Impedance Control	Full Strength
High Temperature Self-Refresh Rate Enable	Disable
DCC Enable	Disable
Partial Array Self Refresh	Full Array

Table 4.5: DDR2 SDRAM Initialization Settings

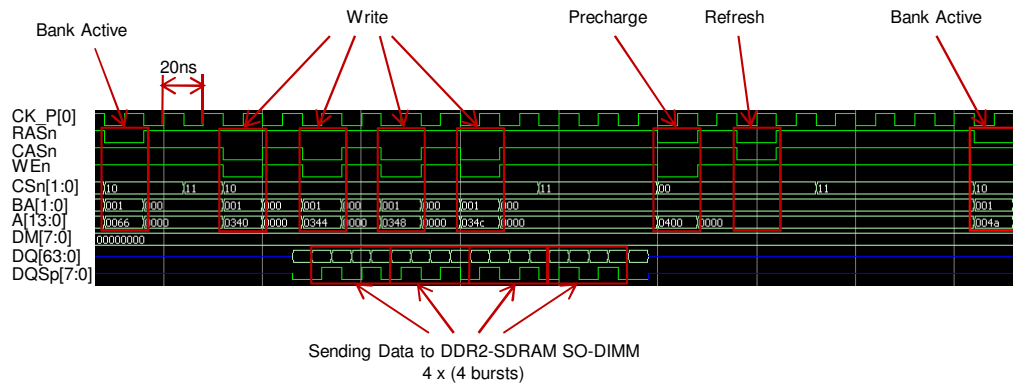


Figure 4.13: Timing Chart of Sequential Write

SDRAM controller sends four sets of *write*, that is, 16 bursts of 64-bit data are sent to the DDR2 SDRAM. DRAM_CLK90 is used for flipping strobe signals (DQSp[7:0]) in the middle of data (DQ[63:0]). It should be noted that there is no interval between data transfer during “sequential read/write”(see DQ[63:0] and DQSp[7:0] in Fig.4.13). The time period needed for the single sequence of “bank open” is represented as

$$T_{singlebankopen} = 280 + 40 \times n_{read/write} \text{ns}, \quad (4.1)$$

where $n_{read/write}$ is the number of *read/write* during the single sequence of “bank open”. For instance, in Fig. 4.13, $280 + 40 \times 4 = 440\text{ns}$ is needed for the sequence. Since 256-bit data is read/written by the single *read/write*, the bandwidth of the sequence is represented as

$$B_{singlebankopen} = \frac{256 \times n_{read/write}}{280 + 40 \times n_{read/write}} \text{Gbps}. \quad (4.2)$$

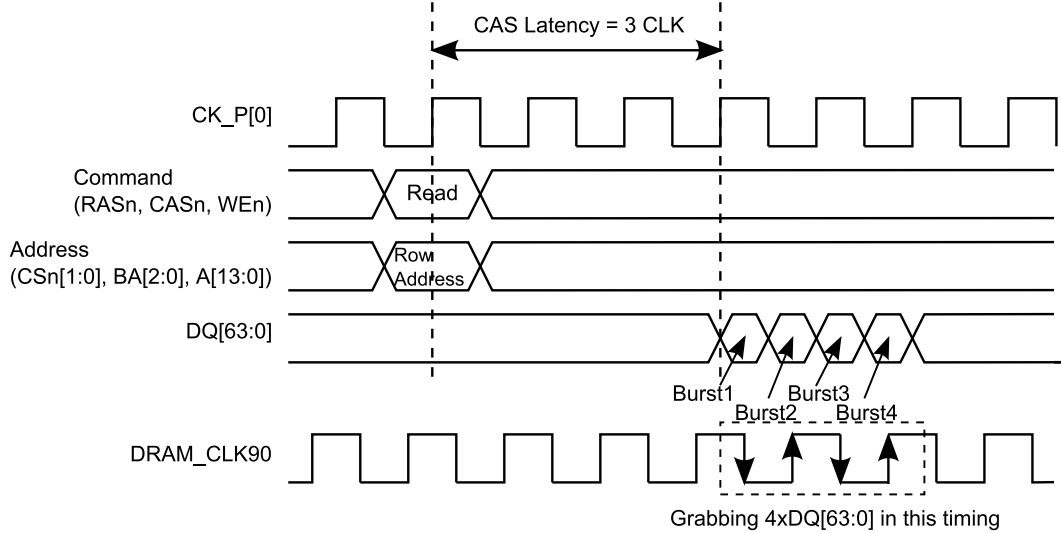


Figure 4.14: Timing Chart of Read

Fig. 4.14 shows how the DDR2 SDRAM controller grabs the data from the DDR2 SDRAM. To grab the data, we flip the data by DRAM_CLK90, without using data strobe signals (DQSp[7:0]).

Arbiter

In this section, we describe an architecture of the arbiter and the idea of arbitration.

Fig.4.15 shows an architecture of the arbiter. The arbiter consists of three modules: input buffer, output buffer, and main module.

A input buffer includes a $256\text{-bits} \times 128\text{-depth}$ built-in FIFO which temporarily stores the data from the BusExp (= the data from the FEE). A output buffer contains the same size FIFO for temporarily storing the data from the DDR2 SDRAM controller to the

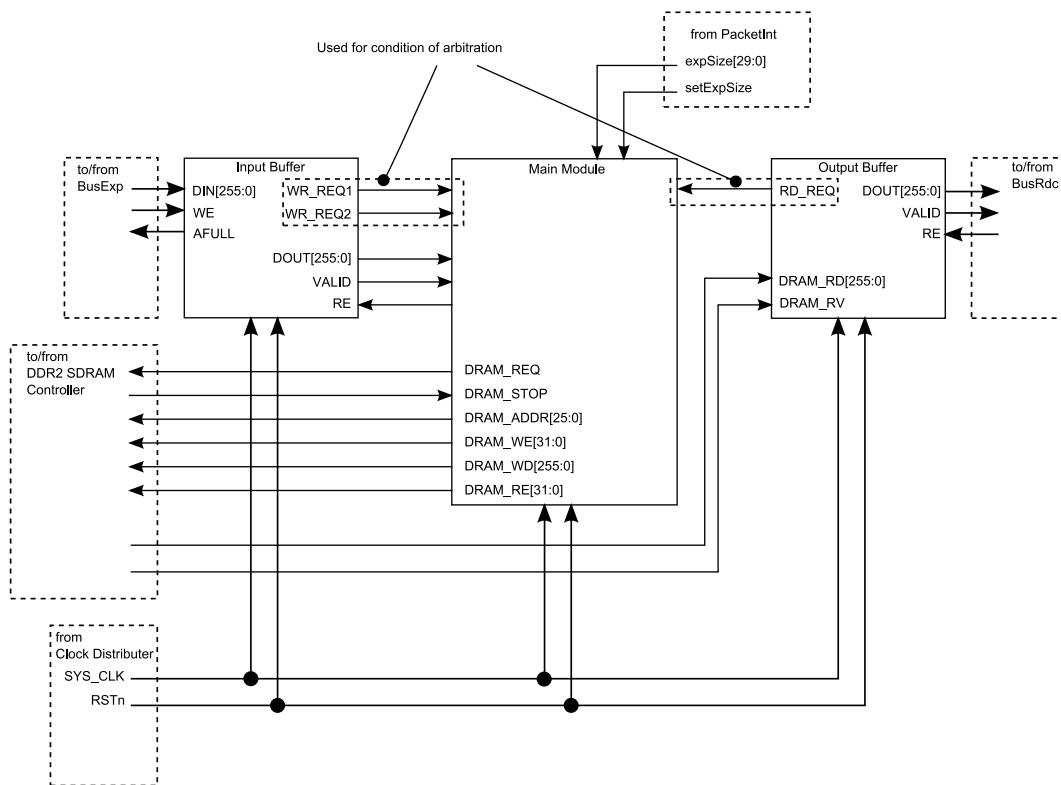


Figure 4.15: Architecture of the Arbiter

BusRdc (= the data to the DAQ system). Each buffer has signals used for arbitration; WR_REQ1, WR_REQ2 of the input buffer and RD_REQ of the output buffer. The former two are the write demand from the FEE side, while the latter one is the read demand from the DAQ system side. Details of these signals will be described later with the idea of arbitration.

A main module makes arbitration and controls the DDR2 SDRAM controller, based on the information from WR_REQ1, WR_REQ2, and RD_REQ. The main module arbitrates the write demand (WR_REQ1, WR_REQ2) and the read demand (RD_REQ) and sends the read/write requirement to the DDR2 SDRAM controller. The main module has two address counters for the read requirement and write requirement whose initial value is 0x000000. The value of the counter for the read/write requirement is sent to the DDR2 SDRAM controller through DRAM_ADDR[25:0] when the read/write requirement is sent. The counter for the read/write requirement is incremented by one everytime read/write requirement is sent. When the end data of a exposure is written/read into/from the DDR2 SDRAM, the counter returns to 0x000000, referring expSize[29:0].

Here we describe the idea of the arbitration. We used a pipeline process to realize the required throughput. The data transfer between the FEE and GESiCA, the access to the DDR2 SDRAM, and the data transfer between GESiCA and the DAQ system are in the pipeline. The main module sends 4 sets of read/write requirements to the DDR2 SDRAM controller after the single arbitration, that is, 1024-bit (4×256 bits) data is read/written from/to the DDR2 SDRAM after the single arbitration. The reason is as follows. Fig. 4.16 will help your understanding.

First, let us consider the data transfer between the FEE and GESiCA, which we define as DT_{in} . The bandwidth of the DT_{in} is expected to be 0.8 Gbps and its bus width is 16 bits, so the data transfer rate of the DT_{in} is 50 MHz. Thus the time period for sending 1024-bit data is $\frac{1024}{16} \times \frac{1}{50\text{MHz}} = 1280\text{ns}$.

Second, consider the data transfer between GESiCA and the DAQ system defined as DT_{out} . The bandwidth of the DT_{out} is 1 Gbps and its bus width is 8 bits. Thus the data transfer rate of the DT_{out} is 125 MHz. In this case, the time period for sending 1024-bit data is $\frac{1024}{8} \times \frac{1}{125\text{MHz}} = 1024\text{ns}$.

Third, consider the data transfer between the FPGA and DDR2 SDRAM, which we define as DT_{ddr2} . From equation (4.1), the DDR2 SDRAM controller needs 440 ns for reading/writing 1024-bit data from/to the DDR2 SDRAM, and the arbiter needs for $5\text{CLK} \times \frac{1}{130\text{MHz}} \sim 40\text{ns}$ for arbitration and sending the read/write requirement to the DDR2 SDRAM controller. Thus the total time period for reading/writing 1024-bit data from/to the DDR2 SDRAM is ~ 480 ns.

Taking into account the difference of these time periods, if the pipelined process as shown in Fig. 4.16 is realized with arbitrating the write demand and the read demand, the data transfer rate from the FEE to GESiCA is limited by the slowest part (in this case DT_{in}). Namely, while DT_{in} and DT_{out} are carried out, the access to the DDR2 SDRAM, DT_{ddr2} , is completed.

Now we discuss how we arbitrate the write demand and the read demand. WR_REQ1 is asserted when 4 or more words (= more than 1024-bit data) are stored in the input buffer. This is the demand to write the image data into the DDR2 SDRAM. RD_REQ, the

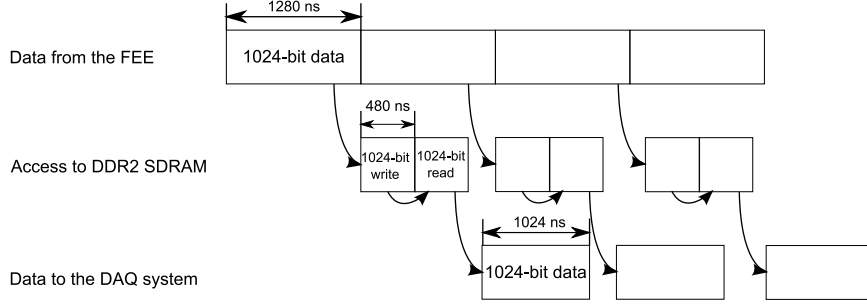


Figure 4.16: An Example of Arbitration Timing

demand to read the image data from the DDR2 SDRAM, is asserted until the number of vacant words in the output buffer becomes 64 or less. The reason why we set the threshold to 64 words is that even if all words of the 32-word input FIFO in the DDR2 SDRAM are filled with the read requirement and all words of the 32-word output FIFO in the DDR2 SDRAM are filled with the read data, we allow the output buffer to store all the words.

When DRAM.STOP is low level, that is, the input FIFO in the DDR2 SDRAM controller affords to accept data, the main module carries out the arbitration. When both of WR_REQ1 and RD_REQ are negated, the main module carries out no operation. When only WR_REQ1 is asserted, the main module sends 4 sets of write requirement to the DDR2 SDRAM Controller in order to write 1024-bit data. When only RD_REQ is asserted, the main module sends 4 sets of read requirement to the DDR2 SDRAM controller in order to read 1024-bit data. When WR_REQ1 and RD_REQ compete against each other, the main module sends 4 sets of read/write requirement in alternate shifts, that is, if the previous requirement sent to the DDR2 SDRAM controller is read, the main module sends a write requirement and vice versa.

The above arbitration is not enough to completely avoid the possibility that the image data is lost. We introduced WR_REQ2, asserted when only 32 or less words can be stored in the output buffer. This assertion acts to give priority to write requirement rather than read requirement. In this case, the main module ignores RD_REQ and continues to send 4 sets of write requirements to the DDR2 SDRAM controller until WR_REQ2 is negated. When the write requirements continue, the DDR2 SDRAM controller sends *write* to DDR2 SDRAM continuously during the single “bank open”. From equation (4.2), the data transfer rate becomes very fast ($B_{singlebankopen} = \frac{256 \times n_{read/write}}{280 + 40 \times n_{read/write}} \rightarrow 6.4\text{Gbps}$ if $n_{read/write}$ is very large). This treatment prevents image data from being lost.

Here we summarize the arbitration.

- WR_REQ1='high', RD_REQ='high'
 - WR_REQ2='high'
 - 4 sets of write
 - WR_REQ2='low'

- * previous command is write
4 sets of read
- * previous command is write
4 sets of write
- WR_REQ1='high', RD_REQ='low'
4 sets of write
- WR_REQ1='low', RD_REQ='high'
4 sets of read
- WR_REQ1='low', RD_REQ='low'
No operation

The size of an exposure may not be multiples of $256 \times 4 = 1024$ bits. In those cases, the main module sends the one/two/three write/read command(s) at the end of an exposure even when there are less than four 256-bit data in the input buffer.

4.4 Performance of Readout Module for Back-End Electronics

In this section, we report the performance of GESiCA itself.

4.4.1 Throughput

Setup

Fig.4.17 shows a block diagram of a setup of our throughput measurement of GESiCA, and Fig.4.18 is a picture of a test system used in the throughput measurement. The test system consists of three parts: GESiCA, a test data generator, and a CMC test board. Since an FEE for HSC is now under development, we developed a test data generator using another GESiCA in order to emulate an ADC in the FEE. The test data generation circuit is implemented in the FPGA on the test data generator. The CMC test board has CMC female terminals with which GESiCA and the test data generator are connected. A DIN96 connector of this test board is connected to a VME crate in order to supply power. A 12-V and a 5-V power are needed. The 12-V power is lowered to 3.3-V by a regulator in the test board. The test board distributes the 5-V and 3.3-V power to both GESiCA and the test data generator. To receive the test data at Gigabit rate, a high-performance DAQ PC is needed. In this test, we used a Dell Poweredge 2900 Quad Core Xeon E5410 (2.33GHz 2x6MB L2 Cache, 1333MHzFSB) running CentOS 5 64 bit. This DAQ PC is connected to GESiCA with a crossover cable. GESiCA and the DAQ PC act as a TCP server and a client respectively.

To start the test, a user should let the DAQ PC send a command to GESiCA. After receiving the command, the readout controller in GESiCA sends a test start command to the test data generator through the CMC connectors. Once the test data generator

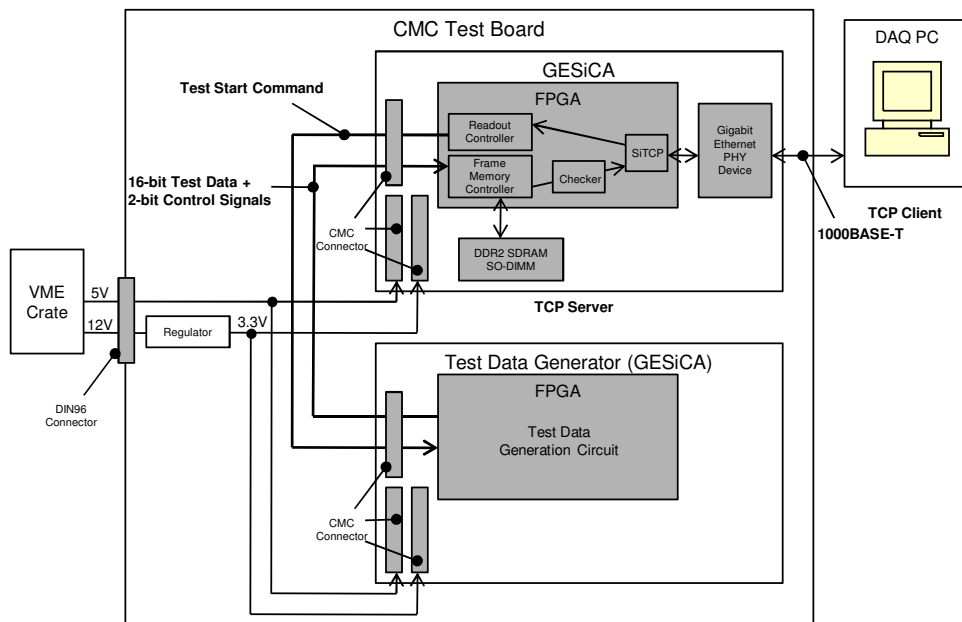


Figure 4.17: Setup for Throughput Measurement

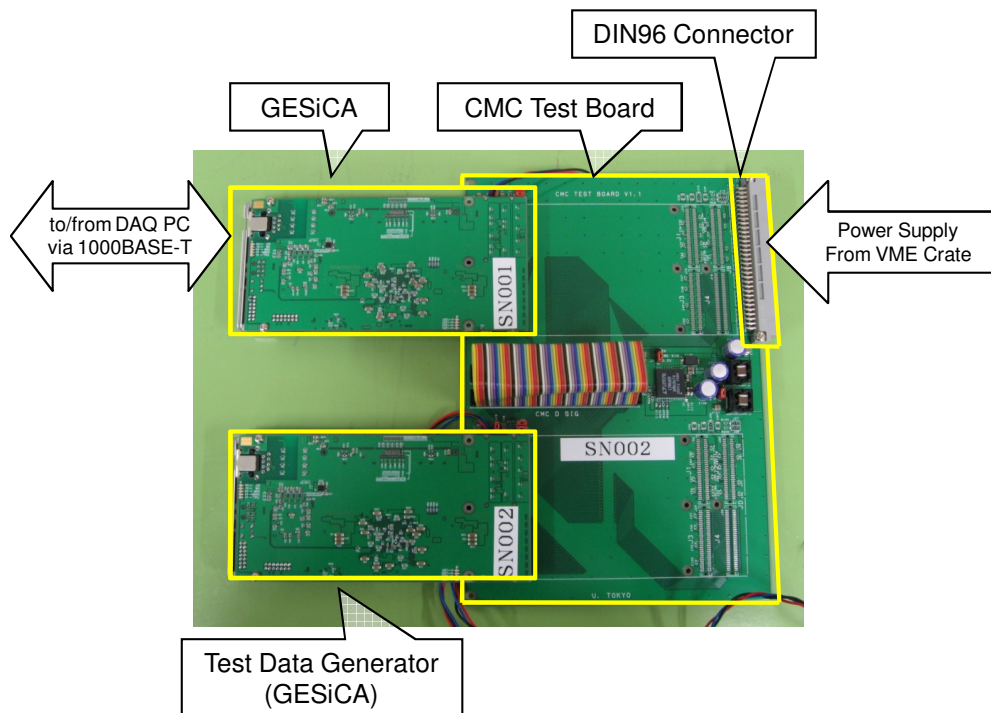


Figure 4.18: Picture of Test System Used in Throughput Measurement

receives the command, it sends 16-bit count-up test data and 2-bit control signals to GESiCA via the CMC connectors. The signal standard of these signals is CMOS 3.3V. The timing of these test signals are same as Fig.4.9 which is expected to be the same timing as the HSC system. The test data generator stops after sending 2-Gbyte test data. The test data comes into the frame memory controller and then is temporarily stored in a DDR2 SDRAM SO-DIMM. The test data are read from the SDRAM and sent to the DAQ PC through Gigabit Ethernet 1000 BASE-T. We put a checker between the frame memory controller in GESiCA and the SiTCP to determine whether the test data is sent to the SiTCP without error. The transfer speed of the test data at the output of the test data generator, which we call the input rate, can be set to various values.

Measurements and Results

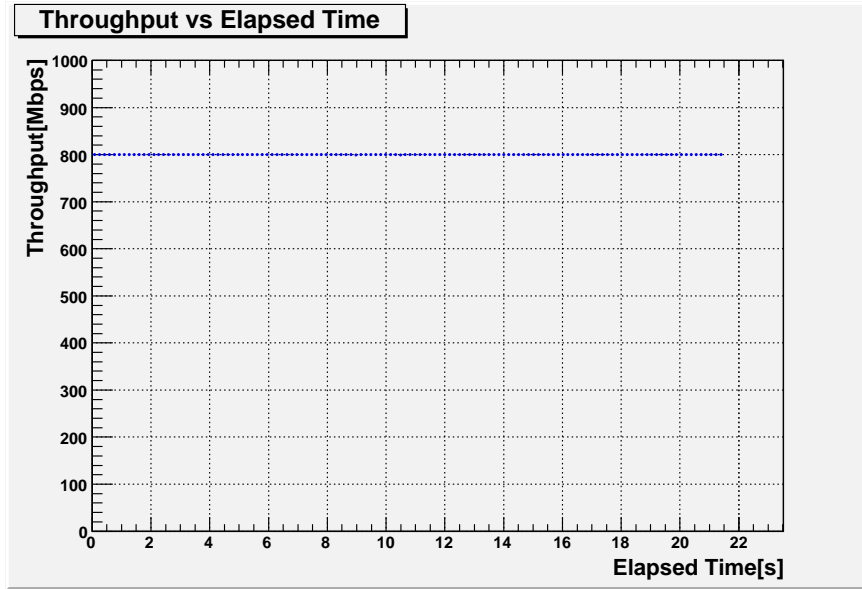


Figure 4.19: Throughput Measurement: Throughput Time Dependence

Two measurements were carried out in this test. First, we fixed the input rate to 0.8 Gbps which is equal to the requirement for the throughput. A simple socket software to measure the throughput was developed, which pushes the time-stamp everytime 16.77Mbyte of data is transferred from GESiCA to the DAQ PC. There is no special configuration for the socket. We then calculated throughput from the time-stamps. The result is shown in Fig.4.19. The X-axis is elapsed time and Y-axis the measured throughput. The throughput from GESiCA to the DAQ PC is 0.8 Gbps, which is same as the input rate. The result satisfies the requirement for the throughput. Note that the measured throughput is stable throughout the readout process.

Second, we measured the throughput of GESiCA against different values of the input rate. We measured a transfer period which GESiCA spent to send whole 2-Gbyte test data

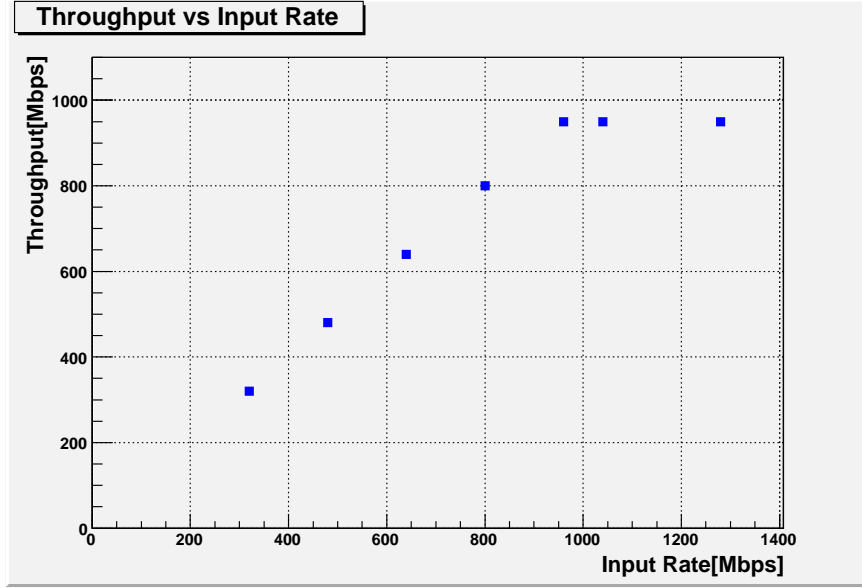


Figure 4.20: Throughput Measurement: Throughput vs Input Rate

to the DAQ PC. From the transfer period we then calculated the throughput. 50 measurements were made for each choice of input rate. We plotted means of the throughput with their errors (root mean square) against the input rate in Fig.4.20. When the input rate is lower than the requirement (0.8 Gbps), the throughput is limited by the input rate. Even when the input rate is faster than 0.8 Gbps, the throughput keeps the speed limited by the input rate up to the theoretical limit of Gigabit Ethernet. The mean of measured throughput is 949.25 Mbps for the 1020-Mbps input and 949.22 Mbps for the 1280-Mbps input. The theoretical limit is calculated by

$$T_{max} = 1\text{Gbps} \times \frac{MSS}{IPG + H_{MAC} + H_{IP} + H_{TCP} + MSS} \quad (4.3)$$

$$= 949.28\text{Mbps} \quad (4.4)$$

Here, MSS is the maximum segment size of TCP, 1460 bytes; IPG is the inter-packet gap defined by the specification IEEE802.3, 12 bytes; H_{MAC} is the overhead length of Ethernet, 26 bytes; H_{IP} is the length of the IP header, 20 bytes; and H_{TCP} is the length of the TCP header, 20 bytes. Thanks to the arbitration circuit in the frame memory controller, a throughput almost equal to the theoretical value was realized. Furthermore, the errors are so small that they cannot be seen on the graph, which indicates that the throughput of GESiCA is stable.

Throughout this test, we observed no error in the checker and we confirmed this readout module has sufficient data transfer accuracy.

4.4.2 Power Consumption

Setup

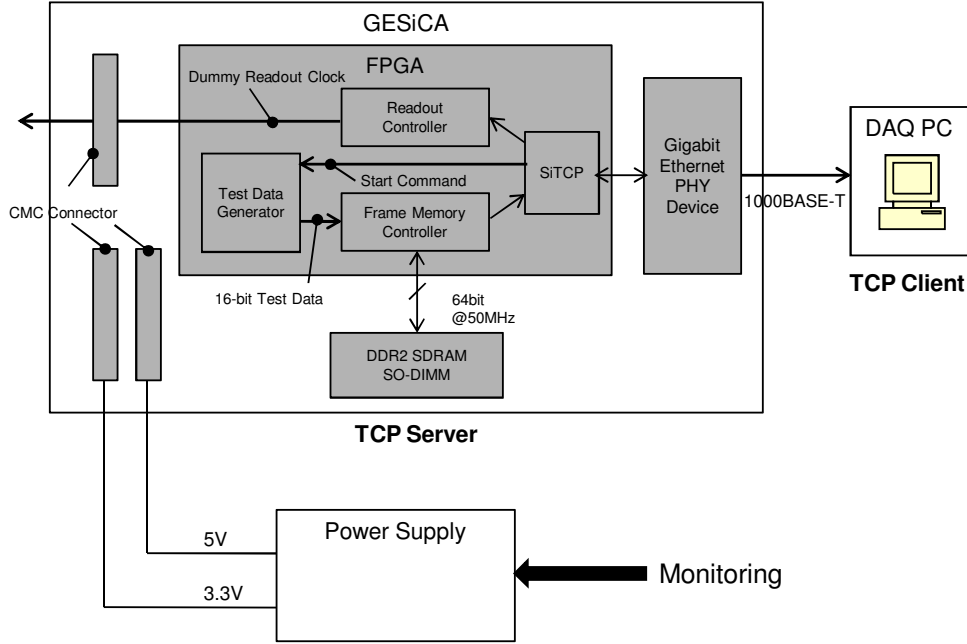


Figure 4.21: Setup of Power Consumption Measurement

Fig.4.21 shows a setup of our power consumption measurement. To estimate power consumption of GESiCA, we connected a power supply to GESiCA by soldering copper lines to the CMC connectors, and monitored the current and the voltage shown on the power supply every 2 seconds. Then we calculated the power consumption by product of the measured current and the measured voltage.

It is expected that the power consumption during operation (reading out CCDs) is much larger than that during standby (no operation to readout). To imitate a state of readout, we operated GESiCA as follows:

1. Put a test data generator in the FPGA and send 16-bit test data to the frame memory controller at the required bandwidth (800Mbps). The test data is stored in the DDR2-SDRAM and then sent to the DAQ PC which is same as the PC used in §4.4.1, via Gigabit Ethernet.
2. Let the readout controller generate dummy readout clock consisting of 32 bits of periodical signals whose cycle is 20 ns while the test data generator sends the test data. This is the fastest clock the readout controller can generate. We estimated the worst case of power consumption by generating the fastest clock.

We started this operation from 3 seconds past the beginning of monitoring.

Results

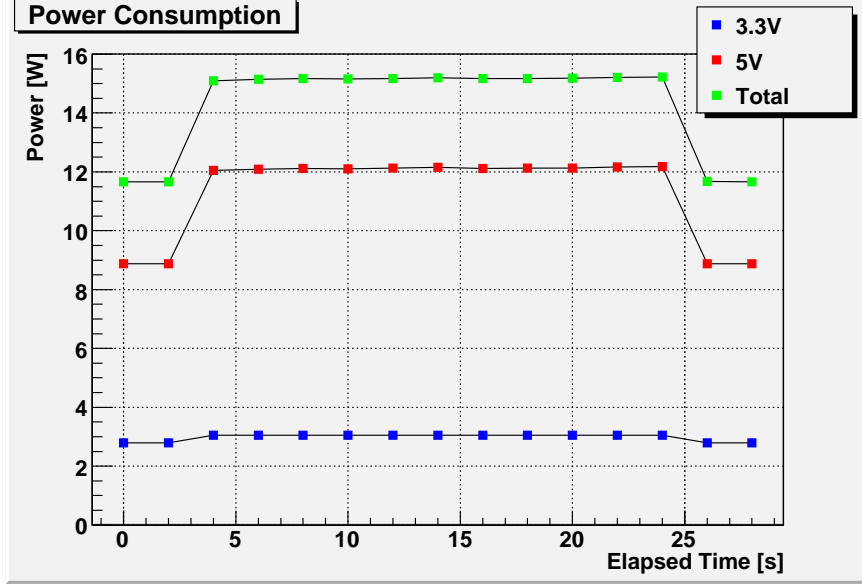


Figure 4.22: Power Consumption Measurement

Fig.4.22 shows the result. The X-axis is the elapsed time and the Y-axis is the the power consumption. Green dots are the total power consumption which is the sum of power consumption of the 3V source (blue) and 5V source (red). There is a large difference between operating state (4s - 24s) and standby (0s-2s, 26s-28s): in total, GESiCA consumes ~ 15 W during operation, while ~ 11.5 W during standby. Note that power consumption is stable during each operation.

4.4.3 Noise Estimation

How GESiCA Causes Noise?

An integration time of a CDS in FEE is determined by the signal from the readout controller. Jitter of the signal causes noise. We estimated the noise by measuring the jitter from the readout controller of GESiCA.

Let us discuss how the jitter causes the noise. Since an integration circuit in a CDS integrates constant current I_{in} for an integration period T , an output of the integration circuit V_{out} is written as

$$V_{out} = \frac{-1}{C} \int_0^T I_{in} dt = \frac{-T}{RC} V_{in} \equiv -\alpha T V_{in}, \quad (4.5)$$

where C, R is capacitance and resistance, respectively. Note that

$$g \equiv \alpha T \quad (4.6)$$

can be regarded as the “gain” of the integration circuit. T actually has jitter δT which defined as the root mean square of T . From equation (4.5), δT causes noise as

$$\delta V_{out} = \alpha \delta T |V_{in}|. \quad (4.7)$$

Here $\alpha \delta T$ can be regarded as a fluctuation of the “gain”. Using a mean of integration period \bar{T} , a mean of the output is written as $\bar{V}_{out} = -\alpha \bar{T} V_{in}$. Thus equation (4.7) becomes

$$\delta V_{out} = \frac{\delta T}{\bar{T}} |\bar{V}_{out}|. \quad (4.8)$$

A CDS has two integration circuits and outputs the difference of the integrated signals. Under the assumption that the readout clock is Gaussian, noise of an output from a CDS caused by the jitter is described as follows:

$$N_{CDS} = \sqrt{\bar{V}_{out,1}^2 + \bar{V}_{out,2}^2} \frac{\delta T}{\bar{T}}. \quad (4.9)$$

where $\bar{V}_{out,1}$ and $\bar{V}_{out,2}$ are the output of two integration circuit in the CDS.

Setup

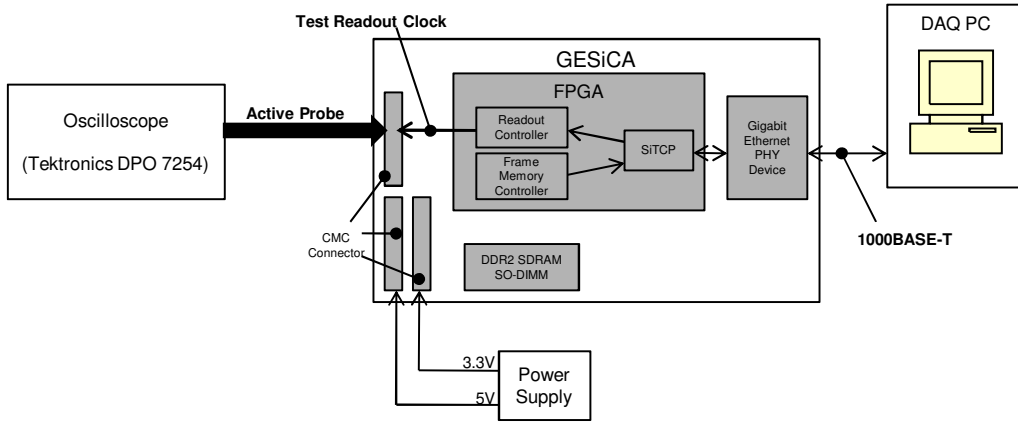


Figure 4.23: Setup of the Jitter Measurement

To estimate noise from jitter, we directly measured jitter of the readout clock from the readout controller of GESiCA. Fig.4.23 shows a setup of our jitter measurement. A digital sampling oscilloscope, Tektronics DPO 7254, with 40-GHz sampling speed and 250-fs timing accuracy and the software TDSJIT3 installed in the oscilloscope are used in order to analyze the jitter. We sampled the readout clock by pressing an active probe on the CMC connector.

The jitter for various integration periods was measured. The integration period is varied by configuring various readout clock patterns by the DAQ PC

Measurement and Estimation

Fig.4.24 plots the result of the measurement. The Y-axis is the rms jitter plotted in seconds, while the X-axis is the integration time. Fig.4.25 shows relative accuracy, namely the ratio of the rms jitter to the integration time, in parts per million (ppm). Errors in these plots come from the measurement accuracy announced in the data sheet of TDSJIT3[31]. The typical integration time is expected to be $\sim 2 \mu\text{s}$ which corresponds to gain ~ 1 in the HSC system. In this case, the jitter of the signal from GESiCA is ~ 10 ppm.

As we have shown in §4.1, the requirement for noise is that noise from the electronics should be much lower than that from a CCD ($\sim 4 \text{ e}^-$) around a bias voltage. The bias voltage is expected to be set to the voltage corresponding to the output of 1,000 ADC counts. Therefore, using the result of the jitter (~ 10 ppm) and equation (4.9), and assuming the jitter is Gaussian, the noise in the few electron region is estimated to be

$$N \sim \sqrt{(1000)^2 + (1000)^2} \times 10 \times 10^{-6} = 0.014 \text{ ADCcounts}. \quad (4.10)$$

The fullwell capacity of a CCD is $\sim 200,000 \text{ e}^-$ and 16-bit ADCs are employed in the FEE. Typical conversion factor is $200,000/2^{16} \sim 3 \text{ e}^-/\text{ADCcount}$. Hence the equivalent noise charge around the few electron region is

$$N_{equiv} = 0.014 \times 3 \sim 0.05 \text{ e}^-. \quad (4.11)$$

Thus we concluded that the noise caused by GESiCA is much lower than the noise of the CCD output, that is, that the requirement is satisfied.

Note that the relative accuracy of integration time $1 \mu\text{s}$ is also about 10 ppm. This means that a higher speed readout is possible in future updates.

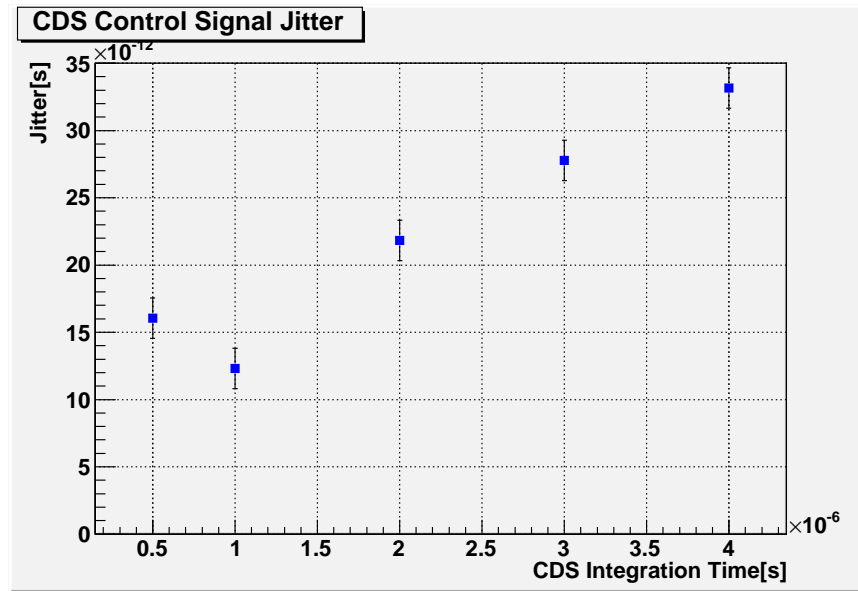


Figure 4.24: Jitter Measurement of Readout Clock. Y-axis is the jitter in seconds.

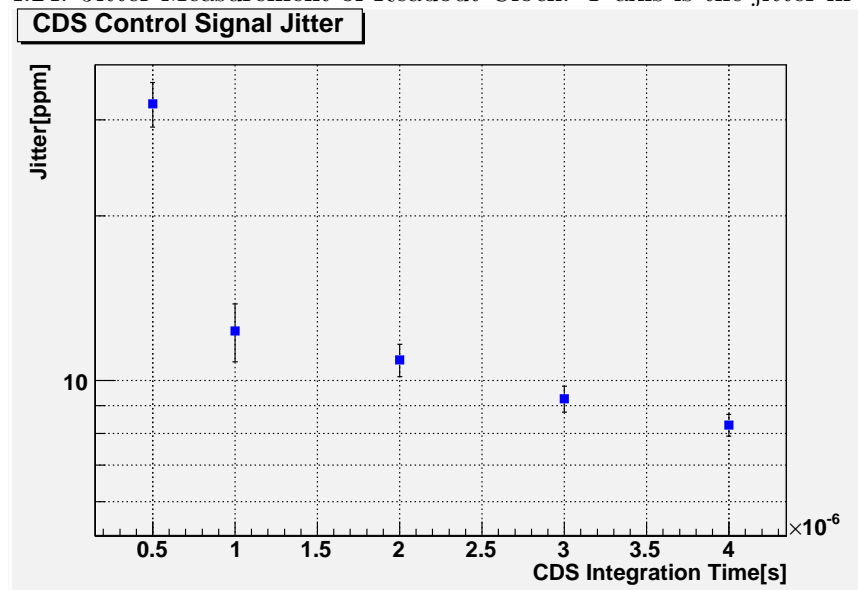


Figure 4.25: Jitter Measurement of Readout Clock. Data is same as Fig.4.24, but Y-axis is the relative accuracy in ppm.

Chapter 5

Prototype Readout System for Hyper Suprime-Cam

We have developed the prototype readout system incorporating GESiCA. The main aim of this system is to estimate the behavior of the whole HSC system. Since the FEE for HSC is under development, we used the FEE of the existing Suprime-cam instead. We checked whether GESiCA can control the FEE, observed behavior of the system, and estimated the influence from GESiCA.

5.1 Prototype Readout System

Fig. 5.1 is a picture of the prototype readout system. This system consists of three 3U Eurocard-size boards (MF2_PWR, MF2_SIG, I/F board), GESiCA, and one backplane. MF2_PWR, MF2_SIG, and the backplane are parts of the FEE of the existing Suprime-Cam called MFront2[15]. A user can control MFront2 by sending commands from a DAQ PC over RS485[32]. The backplane connects three Eurocard-size boards through DIN96 connectors.

Fig. 5.2 shows a schematic of this system. Let us explain the details of each part of the system and the flow of operation.

MF2_PWR is a power management board. This board needs 15-V and ± 5 -V power supplies. Its allowable current is tuned to the Suprime-Cam system and so is not enough for GESiCA. We solved this issue by using another power supply for GESiCA and the I/F board. Another functionality of MF2_PWR is receiving commands for MFront2 from a DAQ PC via RS485 and distributing them to other MFront2 boards.

MF2_SIG is a signal processing card. MF2_SIG can mount up to 4 daughter boards. The single daughter mounts 2 channels. Each channel has one CDS, one 16bit-DAC, and one 16bit-ADC. The single channel receives a signal from one output of CCD and processes the signal in the same way as described in §3.1. Following a readout clock from GESiCA, MF2_SIG controls a CDS and an ADC of each channel on daughter boards with SPI. The readout clock is divided by MF2_SIG and distributed to each channel. Several sets of the digitized 16-bit serial data from the output of each channel are arranged by MF2_SIG into

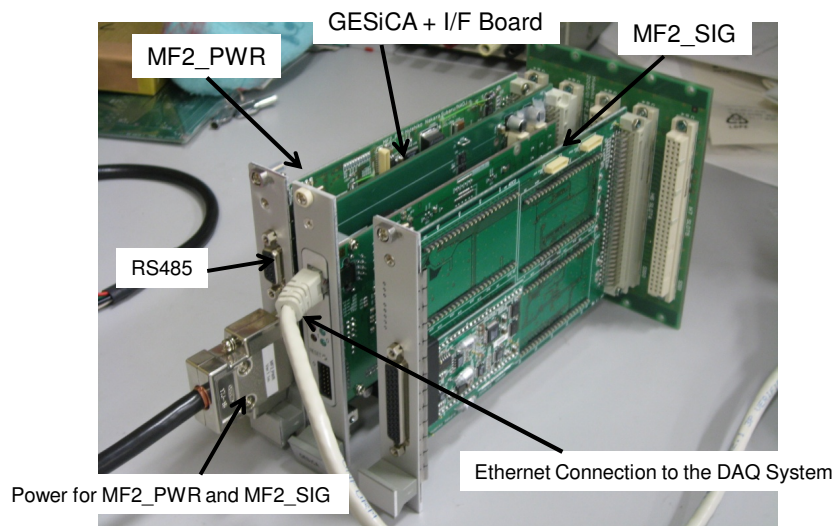


Figure 5.1: Picture of the Prototype Readout System

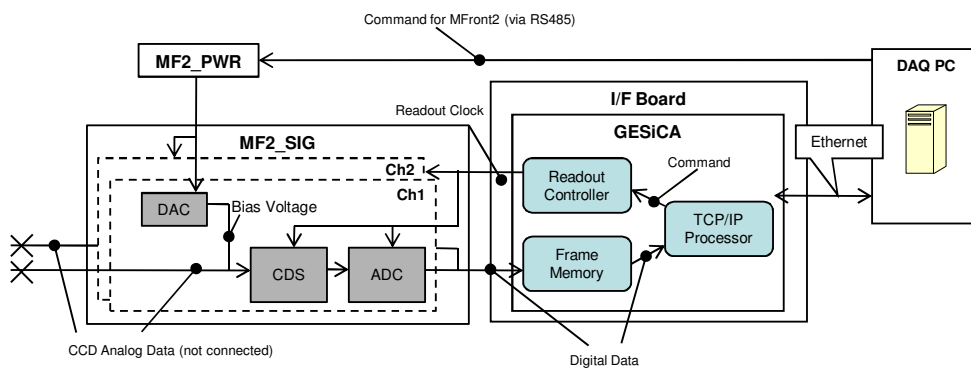


Figure 5.2: Schematic of Prototype Readout System

the single set of 16-bit parallel data and sent to GESiCA with a strobe signal and a valid signal as shown in Fig. 4.9. The range of DAC output (the bias voltage of CCD signal) is from -3V to 3V and it can be controlled by a DAQ PC through MF2_PWR. The input range of the ADC is from 0V to 3V. In this system, MF2_SIG mounts the single daughter board. This system has 2 channels in total.

I/F board is a newly developed board by The University of Tokyo and NAOJ that plays a role of the interface between GESiCA and MFront2. Fig. 5.3 is a picture of the I/F board. This board mounts GESiCA through CMC connectors and converts CMOS 3.3V/LVDS signals to LVDS/CMOS 3.3V by LVDS drivers/receivers. As we mentioned above, the power for the I/F board and GESiCA is supplied not by MF2_PWR but by another power supply directly. The I/F board and GESiCA need a ± 5 -V and a 3.3-V power respectively.

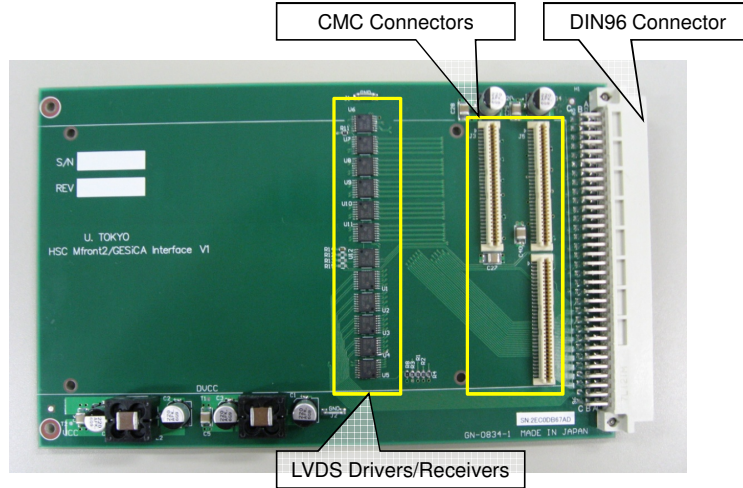


Figure 5.3: Interface Board for GESiCA

The data processing sequence for single pixel data consists of two integrations for the pedestal part and the data part from the CCD signal, subtraction of these two integrated signals, digitization of the analog data from this subtraction, and transfer of the digitized data from an ADC to GESiCA. The former two are operated in a CDS (see Fig.3.2) and the latter two are in an ADC. These operations are controlled by the readout controller of GESiCA. A user can set the timing of these operations by configuring the readout clock. After the digitized data is sent to GESiCA, it is stored in the frame memory temporarily and then sent to the DAQ PC via Ethernet.

5.2 Performance of Prototype Readout System

We evaluated this system by analyzing the output of the FEE acquired by the DAQ PC through GESiCA. Performances we tested are as follows:

1. Time Dependence of the output of the FEE

We checked how an output of the FEE depends on time when the data processing sequence is repeated.

2. Temperature Drift of the CDS “Gain”

We measured how the “gain” of the CDS depends on temperature. Since the “gain” is proportional to the integration period controlled by the readout clock from GESiCA (see §4.4.3), it is affected by temperature drift of the readout clock. It is important to check whether this drift causes major noise or not.

3. Noise at fixed temperatures

We measured how noise of an output of the FEE depends on input level at fixed temperatures. This measurement enables us to estimate jitter of the readout clock from GESiCA. We checked whether it is consistent with the jitter measured in §4.4.3.

This system is not connected to any CCDs, so an input to a CDS comes only from a DAC. Although in the case of normal operation two integrations are operated in a CDS during the single data processing sequence, we operated only one integration in this test for simplicity. We set the period of integration to $2\ \mu\text{s}$ and that of the whole single data processing sequence to $7.52\ \mu\text{s}$. Thus the readout speed of one channel is 133 kHz. In the single data processing sequence, inputs to 2 channels are integrated and 2 sets of 16-bit digitized data are sent to GESiCA. Therefore, total readout speed of this system is 266 kHz.

We tested the prototype system for some fixed temperatures between 20 deg C and 0 deg C by putting the whole prototype system into a constant-temperature oven. The 20-degree environment is regarded as room and the 0-degree the summit of Mauna Kea. We will describe results in the following sections.

5.2.1 Time Dependence of an output of the FEE

We measured time dependence of the output from the FEE. We fixed the input to a CDS by setting the output of the DAC to 0 DAC counts which corresponds to -3V and repeated the data processing sequence without intervals. Fig. 5.4 shows the result. We measured the output of four cases: each channel (ch1 and ch2) for two environments (20 deg C and 0 deg C). The X-axis is the elapsed time from the start of readout, while Y is the output in the units of ADC counts which is sent to the DAQ PC through GESiCA. We plotted the first 3000 data points corresponding to 22.56 ms in the elapsed time.

Instability is observed until ~ 4 ms for all cases: the output is decreasing with elapsed time. After ~ 4 ms, although oscillation of the output is observed, the mean value seems to be stable. This tendency is the same in all of the measurements. The instability at the start of run may be caused by change of temperature of capacitor or other parts in CDS.

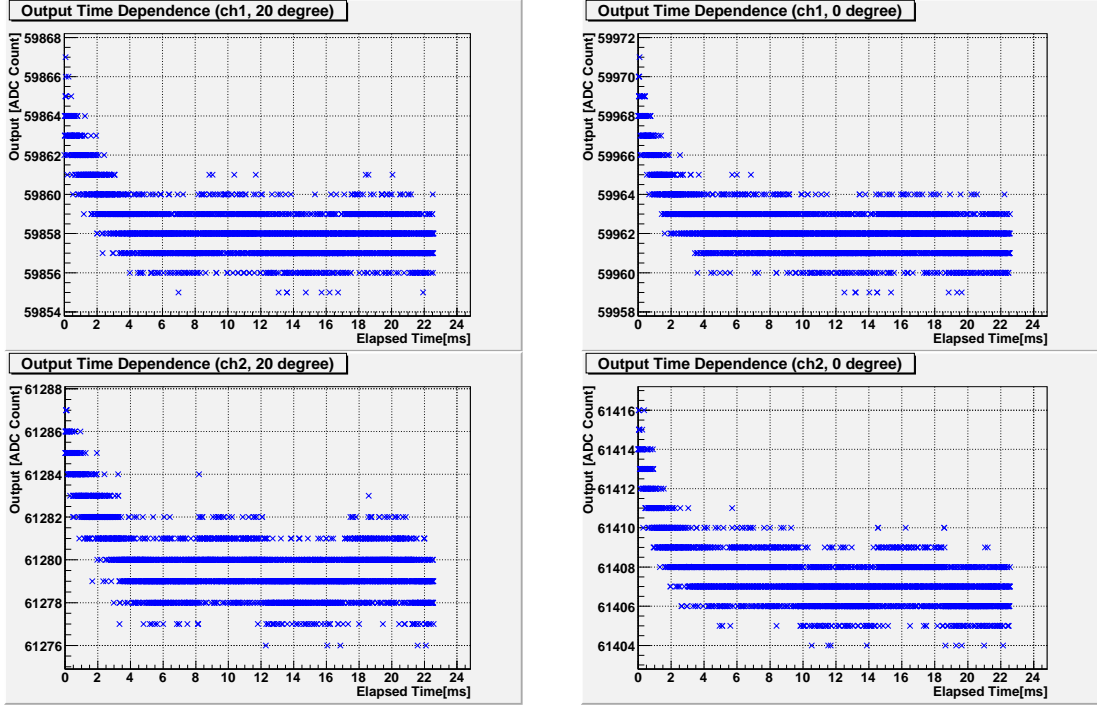


Figure 5.4: Output Time Dependence

Note that this kind of time dependence should be checked in tests of the HSC system in the future. If there is still such a instability in the HSC system, some treatments are needed such as running data processing without CCD inputs at the beginning of a run and then discarding them.

Taking into account this time dependence, in succeeding measurements we use the data after the 2049th sequence (~ 15.4 ms from the start of the run).

5.2.2 Temperature Drift of CDS's gain

Gain

First, we measured the gain of a CDS. We changed input voltage to the CDS by shifting the output of the DAC from 32,768 DAC counts to 0 DAC counts in 512-DAC-count decrements (corresponding to the input to the CDS from 0V to -3V in ~ 0.047 -V decrements). For each voltage, 16384 times of the data processing sequences are repeated. Taking into account the time dependence shown in the previous section, data after 2049th sequence is used, as we mentioned above.

Fig.5.5 is the result for 5 cases of temperature: 20 deg C to 0 deg C in steps of 5 deg C. The X-axis is the input of the CDS and Y the output of the CDS. We took the mean of the data GESiCA acquired as the output. Both of them are converted from ADC counts to voltage. We used the fact that the range of the 16-bit DAC is from -3V to 3V and the 16-bit ADC is from 0V to 3V. We used the rms of the data GESiCA acquired for errors,

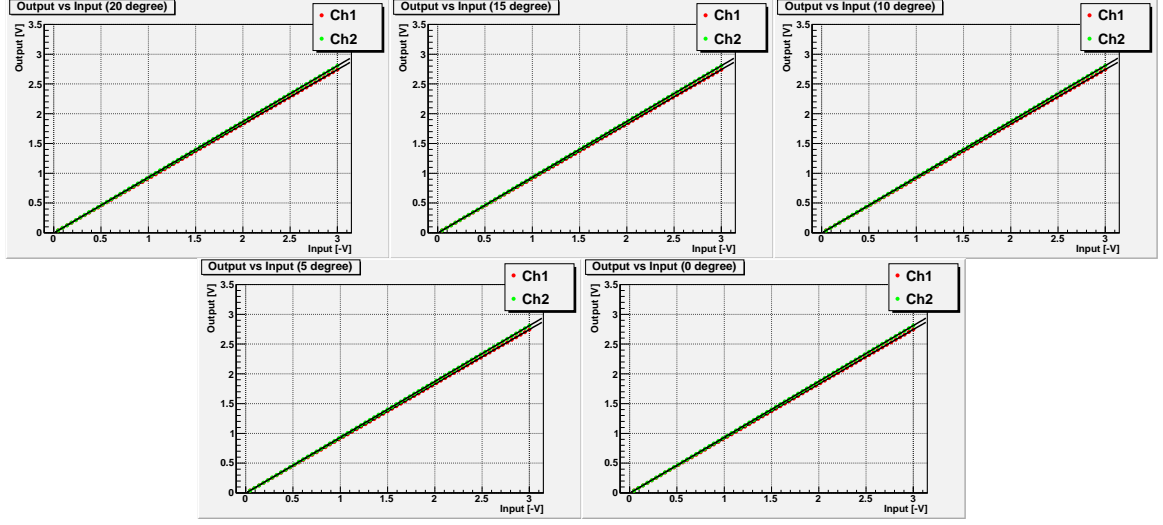


Figure 5.5: Gain of the CDS

but they are too small to be seen.

We estimated the gain by fitting the equation

$$V_{out} = -gV_{in} + h, \quad (5.1)$$

where V_{out} is the output of the CDS, V_{in} is the input of the CDS, g is the gain of the CDS represented as T/RC (see equation (4.5); T is an integration period of the CDS, and R is the resistance and C the capacitance used in the integration circuit in the CDS), and h is the intercept which is ideally zero. Black lines in the Fig.5.5 are fitted lines. The results of the fitting are shown in table 5.1.

There is a difference in gain between ch1 and ch2. Since they use same readout clock from GESiCA, we think this is caused by the difference of individual properties of parts used in the CDS.

Temperature Drift of the Gain

We plot table 5.1 in Fig.5.6 which shows how the gain g depends on temperature. The gain tends to increase when temperature goes down. From table 5.1, we calculated the relative temperature drift of the gain per deg C between two measured points as

$$\Delta_{temp}g = \frac{1}{5} \frac{g_i - g_{i+1}}{(g_i + g_{i+1})/2}, \quad (5.2)$$

where g_i and g_{i+1} are the gain of neighboring temperatures of this measurement. The result is shown in table 5.2.

Discussion

From table 5.2, the relative temperature drift is roughly $\sim 100\text{ppm/C}$. Here we discuss whether the temperature drift causes major noise or not.

channel1		
temperature (deg)	g (Gain)	h
20	0.914449 ± 0.000005	-0.002885 ± 0.000008
15	0.914901 ± 0.000005	-0.002837 ± 0.000008
10	0.915295 ± 0.000005	-0.002778 ± 0.000008
5	0.915661 ± 0.000005	-0.002726 ± 0.000008
0	0.915986 ± 0.000005	-0.002691 ± 0.000008
channel2		
temperature (deg)	g (Gain)	h
20	0.936805 ± 0.000005	-0.004819 ± 0.000008
15	0.937363 ± 0.000005	-0.004840 ± 0.000008
10	0.937872 ± 0.000005	-0.004848 ± 0.000008
5	0.938355 ± 0.000005	-0.004865 ± 0.000008
0	0.938798 ± 0.000005	-0.004895 ± 0.000008

Table 5.1: Gain of the CDS from the Fit

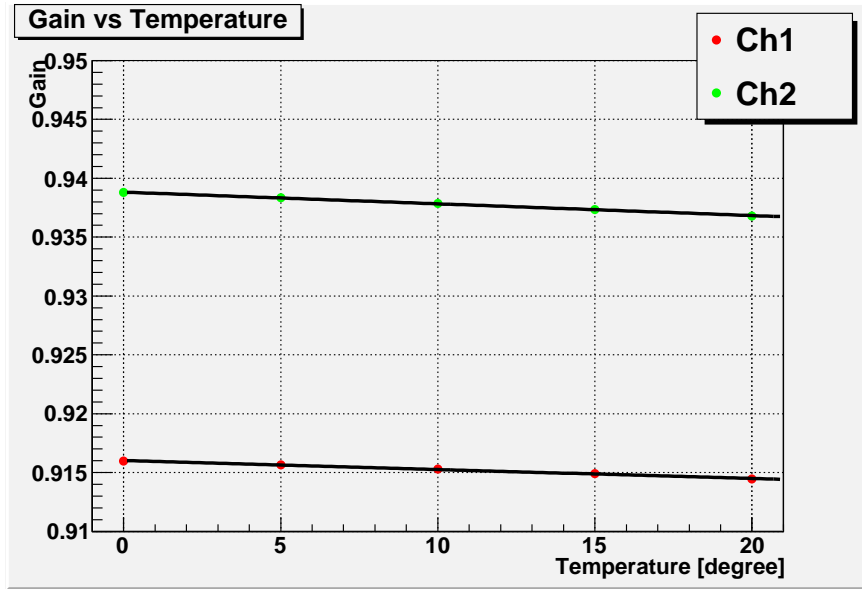


Figure 5.6: Temperature Drift of the Gain of the CDS

channel1		channel2	
temperature (deg)	Δ_{tempg} (ppm/C)	temperature (deg)	Δ_{tempg} (ppm/C)
0-5	-70	0-5	-94
5-10	-79	5-10	-102
10-15	-86	10-15	-109
15-20	-98	15-20	-120

Table 5.2: Relative Temperature Drift

First, let us discuss the worst case of the output's drift resulting from the temperature drift, that is, when the number of electrons from a CCD to the CDS is a maximum. In this case, the ADC outputs almost the maximum value ($\sim 2^{16}$ ADC counts). Hence the temperature drift is

$$2^{16} \times 100 \times 10^{-6} \sim 6.5 \text{ADCcounts/C.} \quad (5.3)$$

Assuming the conversion factor $\sim 3e^-$ in the same way as §4.4.3, this temperature drift corresponds to $\sim 20e^-/\text{C}$. When a CCD is connected to the FEE, this noise is much smaller than the Poisson noise of this case resulting from electrons produced in the CCD (when the fullwell of a CDS is $200,000e^-$, the Poisson noise is $\sqrt{200,000} \sim 450e^-$). Thus, in this case, the noise resulting from the temperature drift is negligible.

Second, let us discuss the influence of the temperature drift when the Poisson noise is small, that is, the output from a CCD is a few electrons. In this case the input to the CDS is almost equal to the bias level set by the DAC. The bias level will be set to 1,000 ADC count(see §4.4.3). The temperature drift around the bias level is roughly

$$1000 \times 100 \times 10^{-6} = 0.1 \text{ADCcounts/C.} \quad (5.4)$$

Using the conversion factor $\sim 3e^-$, the temperature drift is equivalent to $\sim 0.3e^-/\text{C}$, which is much lower than CCD readout noise corresponding to $\sim 4e^-$ (Note that the CCD readout noise is different from the Poisson noise). Even if the output from the CCD is a few electron, the noise resulting from the temperature drift is negligible.

The problem of this measurement is that we cannot measure the temperature drift purely caused by GESiCA, since we put the FEE and GESiCA together into a constant-temperature oven. To understand the influence from GESiCA clearly, it is necessary that only GESiCA is put into an oven while MF2 is left outside. According to Nakaya et al.[15] however, the temperature drift of MF2_SIG itself is within a few 100 ppm/C, which is comparable to our result. Thus we conclude at least GESiCA does not make the temperature drift of the system worse. This conclusion is also reasonable considering the fact that the maximum deviation of the frequency of clock source mounted on GESiCA is $\pm 100\text{ppm}$ when the temperature changes from -10 deg C to 70 deg C (see data sheet [33]), which corresponds to $\sim \pm 1.2 \text{ ppm/C}$.

5.2.3 Noise at fixed temperatures

Theoretical Prediction

We measured noise for different values of input signal to the CDS in the MF2.SIG by changing the DAC which provides bias voltage to the input at a fixed temperature. We defined the noise as the root mean square of the output acquired by the DAQ PC through GESiCA. The expected noise is written as

$$N = \sqrt{N_{int}^2 + N_{jitter}^2}, \quad (5.5)$$

where N_{int} is the noise of the output that MF2.SIG intrinsically has and N_{jitter} is the noise of the output caused by jitter of the readout clock to a CDS from GESiCA. Substituting equation (4.9) into equation (5.5), we obtain

$$N = \sqrt{N_{int}^2 + \bar{V}_{out}^2 \left(\frac{\delta T}{\bar{T}} \right)^2}. \quad (5.6)$$

Here, \bar{V}_{out} is the mean output voltage. When \bar{V}_{out} is changed, noise of MF2.SIG N_{int} emerges as the intercept under the assumption that N_{int} does not depend on \bar{V}_{out} , and the jitter in the form of relative accuracy $\frac{\delta T}{\bar{T}}$ emerges as the gradient of the curve.

Measurements and Results

We changed the output voltage of the DAC from 32,768 DAC counts to 0 DAC counts in 512 DAC count decrements (corresponding to the input to the CDS from 0V to -3V in $\sim 0.047V$ decrements). For each voltage, 16384 times of the data processing sequences are repeated. Taking into account the time dependence measured in §5.2.1, the data after the 2049th sequence is used to calculate the mean and rms of the output which is then processed in MF2.SIG and sent to the DAQ PC through GESiCA.

The results are shown in Fig.5.7 and Fig.5.8. The former is the result of 20 deg C and the latter is that of 0 deg C. The outputs are fitted using a Gaussian. The X-axis is the mean of the fitting result, while Y is the sigma of the fitting result. We assumed the same error δy for all of the measured noises conservatively estimated as $\delta y = \sqrt{\frac{\sum_{all data} (y_{i+1} - y_i)^2}{n}}$, where y_i and y_{i+1} are noises of the neighboring measured points and n is the number of data points. The bottom X-axis and the left Y-axis are in units of ADC counts, while the top X-axis and the right Y-axis are in units of electrons produced in a CCD. Here we assumed fullwell capacity of a CCD of $200,000e^-$ and in this case the conversion factor becomes $\sim 3e^-/\text{ADC count}$ in the same way as §4.4.3. Black lines are the fit results of the estimated noise (cf. equation (5.6)). We can estimate N_{int} and T_{jitter} by the fit and their results are shown in table 5.3.

Discussion

Here we consider the source of N_{int} . One of the sources is the ADC; when a constant level is input to the ADC, the rms of the ADC output is ~ 0.2 ADC counts according to

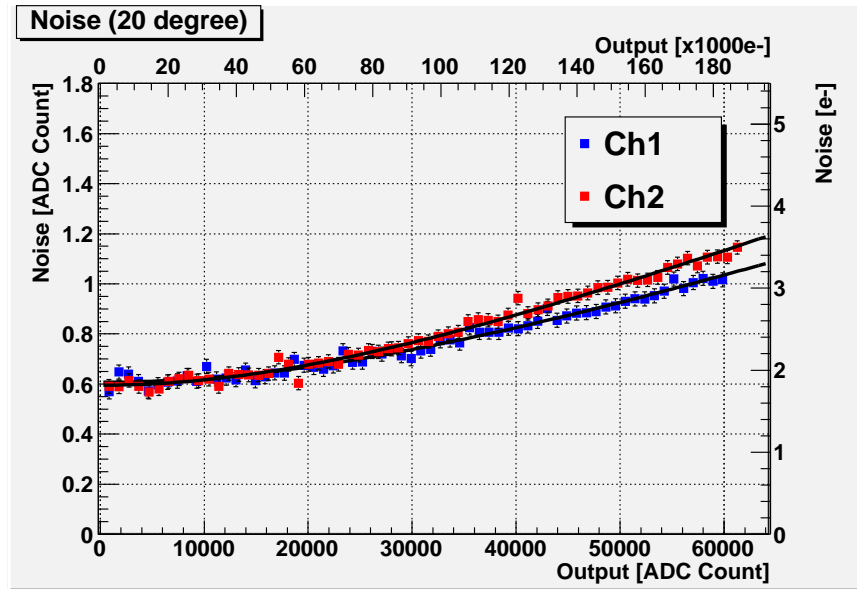


Figure 5.7: Noise of the Output (20 degree C).

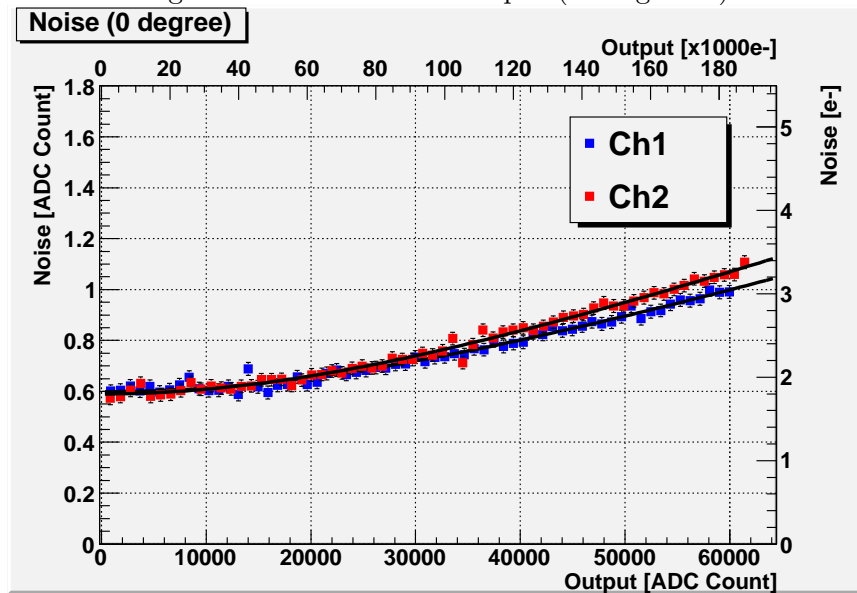


Figure 5.8: Noise of the Output (0 degree C).

temperature (degrees C)	channel	N_{int} (ADC count)	$\delta T/\bar{T}$ (ppm)
20	ch1	0.607 ± 0.006	13.9 ± 0.2
	ch2	0.595 ± 0.006	16.1 ± 0.2
0	ch1	0.598 ± 0.005	13.3 ± 0.2
	ch2	0.589 ± 0.005	14.8 ± 0.2

Table 5.3: Result of Fit

its data sheet [34]. The other source is thought to come from the output of the DAC. The fluctuation of the DAC is mainly caused by that of a reference voltage to the DAC. Stability of the reference voltage is important to reduce N_{int} . Note that N_{int} of the fitting result corresponds to $\sim 1.8e^-$ and it is smaller than the requirement ($4e^-$).

As for the jitter of the readout clock from GESiCA, there is a good agreement between the fit result in table 5.3 and direct measurement described in §4.4.3. From this measurement, the noise from the jitter around the bias level (1000 ADC counts) is $\sim \sqrt{1000^2 + 1000^2} \times 16 \times 10^{-6} \times 3 \sim 0.08e^-$ in the worst case. Here we used equation (4.9). This noise level is much lower than that from MFront2 ($\sim 1.8e^-$). Hence the noise caused by GESiCA is negligible, and same could be said for the whole HSC system.

The jitter tends to become smaller when temperature is going down. The noise from GESiCA is still small in the environment of Mauna Kea (0 deg C). Between ch1 and ch2, there is a difference in jitter to the order of ~ 1 ppm. It would appear that this difference is caused by the individual difference of parts employed in each channel. This difference corresponds to $\sim 0.005e^-$ around the bias level (1000 ADC counts). This difference is sufficiently smaller than the noise from MF2.SIG ($\sim 1.8e^-$). In fact no major difference between ch1 and ch2 can be seen around 1000 ADC counts in Fig. 5.7 and Fig. 5.8.

Chapter 6

Conclusion

6.1 Summary

We have successfully developed the prototype readout module *GESiCA* for the HSC. *GESiCA* is compact (149 mm x 79 mm x 12 mm), light (~ 100 g) and has low power consumption (~ 15 W). The important performance factors of *GESiCA* are throughput and noise. As a result of our performance tests, we confirmed that *GESiCA* can transfer data through a Gigabit Ethernet link not only at the speed requirement (0.8 Gbps) but at close to the theoretical upper limit of 0.949 Gbps. We estimated the noise through the jitter of the readout clock from *GESiCA*. The measured jitter is ~ 10 ppm which corresponds to $\sim 0.5e$, satisfying the requirement for noise.

We developed the prototype readout system which consists of *GESiCA* and the analog front end of the existing Suprime-Cam in order to predict the behavior of the readout system of the HSC. The behavior of the prototype system is understood and it was seen that the noise level caused by *GESiCA* is extremely small. This demonstrates the feasibility of a HSC readout system incorporating *GESiCA*.

6.2 Future Prospect

Although it is confirmed that *GESiCA* satisfies the basic requirements, the following issues remain:

- Replacing 1000BASE-T with 1000BASE-SX
Since the distance between *GESiCA* and the DAQ system in the control room is expected to be of a few 100 m, it is necessary to employ optical communication for Gigabit Ethernet. Although the prototype *GESiCA* employs 1000BASE-T with UTP in order for easy evaluation, it is planned to be upgraded to 1000BASE-SX with optical fiber.
- Synchronization of readout clocks of two *GESiCA*s
We employ two *GESiCA*s for the entire HSC readout system. There remains issues of how to synchronize the readout clocks of two *GESiCA*s.

In addition, other modules for the BEE listed in §3.3 have to be developed. The FEE for the HSC readout system is under development at NAOJ. We plan to finish evaluating the entire system with CCD at the end of 2009.

Appendix A

Details of UDP Communication with GESiCA

A.1 RBCP

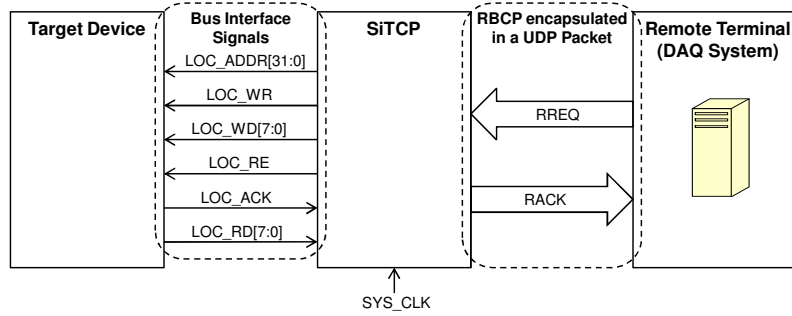


Figure A.1: Schematic of communication over RBCP

The original protocol called remote bus control protocol (RBCP) is used for UDP communication via SiTCP. Fig.A.1 shows a schematic how a remote terminal (DAQ system) holds UDP communication with a target device through the SiTCP. The RBCP accesses to a bus of the target device, based on a request-acknowledge method. After receiving a request packet (RREQ), the SiTCP accesses the bus using bus interface signals and then sent back an acknowledge packet (RACK), which tells a result of the access, to the remote terminal. The bus interface signals are synchronized to SYS_CLK.

Table A.1 shows a packet format of the RBCP. Ver[3:0] and Type[3:0] are reserved for future use and now should be 0xFF. Command[3:0] indicates whether a packet is for reading the bus or writing. Flag[3:0] indicates a result of an access. ID[7:0] is used for identifying a packet, whose detail will be described later. Length[7:0] indicates the length of the data handled in the single packet in units of byte. The range of the Length is from 0x00 to 0xff. Address[31:0] indicates the start-bus address. Data_1[7:0] - Data_N[7:0] is

Position in byte	Name	Description
0	Ver.[3:0]/Type[3:0]	Should be 0xFF, which means the test version.
1	Command[3:0]/Flag[3:0]	Command 0xC = Read operation 0x8 = Write operation Flag [3]: 0=Request, 1=Acknowledge [2:1]: always zero [0]: 0=Normal, 1=Bus Error
2	ID[7:0]	Number to identify, any number can be used
3	Length[7:0]	Length of read/write access
4	Address[31:24]	Bus Address
5	Address[23:16]	Bus Address
6	Address[15:8]	Bus Address
7	Address[7:0]	Bus Address
8	Data_1[7:0]	1st Read/Write data
⋮		
7+ Length	Data_N[7:0]	The last (Length-th) read/write data

Table A.1: Packet Format of RBCP

the data dealt in the packet and N has the same value as indicated by Length[7:0].

We show sequences of writing/reading a bus in the target device as follows.

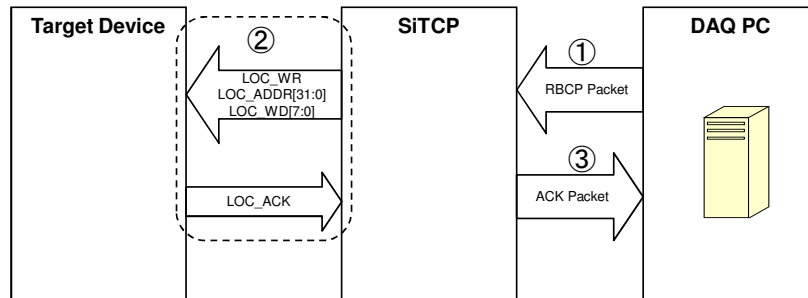


Figure A.2: RBCP Write Sequence

Write Sequence Fig.A.2 shows a sequence of write in the RBCP. To write a bus in the target device, a user send a RREQ with write command (command[3:0]=0x8) with an arbitrary ID[7:0], length of data (Length[7:0]), a top address of the bus (Address[31:0]), and data (Data_1[7:0] - Data_N[7:0]) (① in Fig.A.2). Up to 255 sets of 1-byte data can be handled in the single RBCP packet. After receiving a single RREQ which indicates write,

the SiTCP first sends the top address of the bus (LOC_ADDR[31:0]) and 1-byte top data (LOC_WD[7:0]) with asserting LOC_RW, to the target device. After this, when the target device asserts LOC_ACK, the SiTCP sends the next 1-byte data and the incremented address with assertion of LOC_WR. This process is repeated until all data in the RBCP are sent to the target device (② in Fig,A.2). After all the data is sent, the SiTCP sends a RACK with Flag[3:0]=0x8 which indicates the access succeeded and ID[7:0] same as the above RREQ, to the remote terminal. If LOC_ACK is not sent from the target device for 120 ms before finishing to send all the data, SiTCP sends RACK with Flag[3:0] =0x9 to the remote terminal (③ in Fig,A.2).

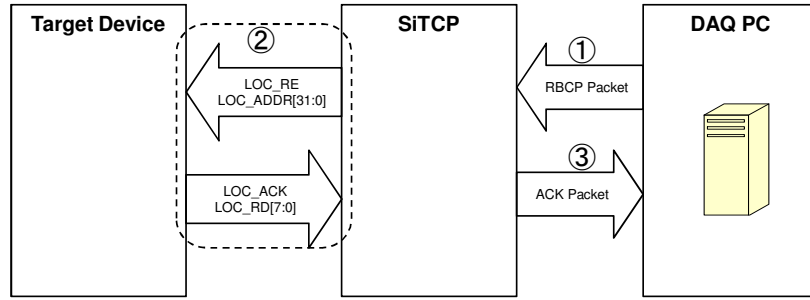


Figure A.3: RBCP Read Sequence

Read Sequence Fig.A.3 shows a sequence of read in the RBCP. A RREQ with read command (command[3:0]=0xC) instructs the SiTCP to read a bus in the target device. In this RREQ, a user should set an arbitrary ID[7:0] and indicate data length (Length[7:0]) and a top address of data (LOC_ADDR[31:0]) which the user wants (① in Fig,A.3). The user can receive up to 255 sets of 1-byte data by the single RREQ. After receiving the single RREQ with read command, the SiTCP sends the top address(LOC_ADDR[31:0]) with assertion of LOC_RE to the target device. After target device sends the data corresponding to the address (LOC_RD[7:0]) with asserting LOC_ACK, the SiTCP sends to the next incremented address to the target device with LOC_RE. This process repeats until all data are sent from the target device (② in Fig,A.3). After all the data is obtained, the SiTCP send a RACK with Flag[3:0]=0x8 which means the bus access was successful and Length[7:0] sets of data. If the SiTCP does not receive a RACK for 120 ms before receiving all the data, it sends a RACK with Flag[3:0]=0x9 which means bus error to the remote terminal.

A.2 RBCP Communication with GESiCA

A user can send a command to GESiCA and configure the readout clock of GESiCA. The RBCP is used for this configuration. Address map for commands and readout clock configuration is shown in table A.2. Detailed description is as follows. §4.3.3 and §4.3.4 will help your understanding.

- Commands

- invoke

Start a *procedure* of readout clock which is memorized in the PrRAM. A user must send this command in a RBCP packet with command 'read'. 'start address' in table A.2 indicates start address of the PrRAM which corresponds to the top of the *procedure*. Return value is 0x00 when the pattern sequencer has successfully started and 0x01 when the pattern sequencer is still busy and the request has been denied.

- reset

Stop the movement of the pattern sequencer and the pattern generator. All output signals of readout clock are set to low level. A user must send this command in a RBCP packet with command 'read'. This command always succeeds and its return value is 0x00.

- busy?

Ask the state of the pattern sequencer. This command should be sent in a RBCP packet with command 'read' and its return value is 0x00 when the pattern sequencer is idle and 0x01 when busy. Note that there is a possibility that the pattern generator is still generating the readout clock even when the return value 0x00 is obtained.

- Configuration

- setExpSize

Set exposure size (ExpSize[29:0]) in order to let the frame memory controller know where a image data header is added. Note that 'ExpSize[29:0]' is equal to "the actual exposure size -1". For example, the exposure size of 1G pixels is corresponding to "ExpSize[29:0]=0x3ffffff". When a user configure the exposure size, a RBCP packet with command 'write' must be sent. Since the size of 'ExpSize' is 30 bits, it is divided into 4 data of the RBCP and then sent to GESiCA. Thus octet# in the address space running from 0x00 - 0x03 is needed (see table A.2). For each octet#, the divided 'ExpSize[29:0]' must be sent to GESiCA (see table A.3). When a user wants to ask a current value of 'ExpSize[29:0]' set in the frame memory controller, a RBCP packet with command 'read' must be sent to GESiCA and then the 'ExpSize[29:0]' divided in the same way as above is sent to a remote terminal.

- PtRAM

Configure the PtRAM in the readout controller. Since the width of the PtRAM is 54 bits, 8 octets are needed to handle the single word of the PtRAM. Table A.4 shows how the single word is divided into each octet. When a RBCP packet with command 'write' is sent to GESiCA, the PtRAM is overwritten, while that with command 'read' is sent, the programmed value of PtRAM is read and the sent to the DAQ system.

– PrRAM

Configure the PrTAM in the readout controller. Since the width of the PrRAM is 72 bits, 16 octets are needed to handle the single word of the PrRAM. Table A.5 shows how the single word is divided into each octet. A RBCP packet with command 'read'/'write' has the same function as PtRAM case.

Acknowledgement

I would like to express my great appreciation to Prof. H. Aihara for his great support and useful suggestions.

I greatly appreciate the help of T. Uchida for his advice on my research from the basics to difficult parts of hardware development. I am grateful to M. Iwasaki, H. Kakuno, T. Abe, N. Hastings for useful advice.

I am indebted to H.Nakaya who is a member of National Astronomical Observatory of Japan (NAOJ) for giving me useful suggestions for developing GESiCA and letting me test GESiCA with the existing Suprime-Cam front-end electronics. I am also thankful to M.Tanaka who is a member of KEK for letting me use a high-quality oscilloscope for the jitter measurement. I am also grateful to M.Takada who is a member of Institute for the Physics and Mathematics of the Universe (IPMU), University of Tokyo for teaching me basics of cosmology and weak lensing.

My appreciation also goes to the students of our group. I would like to appreciate H. Fujimori, for many discussion about development, and S. Mineo, for a nice implementation of the readout controller into GESiCA. I express my appreciation to C. Ng for corrections of English in this thesis. I thank Y. Nakahama, H. Nakayama, S. Sugihara, A. Kusaka, K. Tanabe, K. Yamada for my happy life in Hongo.

Finally, I would like to express my gratitude to my parents and brothers for supporting me both financially and mentally.

Bibliography

- [1] Adam G. Riess *et al*, *Observational evidence from supernovae for an accelerating universe and a cosmological constant*, Astronomical J. 116: 1009-38, 1998
- [2] Adam G. Riess *et al*, *The Farthest Known Supernova: Support for an Accelerating Universe and a Glimpse of the Epoch of Deceleration*, ApJ, 560:49-71, 2001
- [3] E. Komatsu, *et al*, *Five-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation*, ApJ accepted and will be published soon
- [4] M. Bartelmann, *et al*, *Weak Gravitational Lensing*, arXiv:astro-ph/9912508, 1999
- [5] Y. Taniguchi, *et al*, *Modern Astronomy Series 4: Galaxy I*, Nihonhyouronsya , 2007
- [6] M. Takada, *et al*, *Tomography of lensing cross power spectra*, Astrophys.J.601:L1,2004
- [7] D. Huterer, *et al*, *Systematic Errors in Future Weak Lensing Surveys: Requirements and Prospects for Self-Calibration*, Mon.Not.Roy.Astron.Soc. 366 101-114, 2006
- [8] R. Massey, *et al*, *The Shear TEsting Programme 2: Factors affecting high precision weak lensing analyses*, Mon.Not.Roy.Astron.Soc.376:13-38, 2007
- [9] S. Miyazaki *et al*, *HyperSuprime: project overview*, Proc. SPIE, Vol. 6269, 62690B, 2006
- [10] S. Miyazaki *et al*, *Subaru Prime Focus Camera - Suprime-Cam -*, PASJ 54, 833-853, 2002
- [11] S. Miyazaki *et al*, *Searching for Dark Matter Halos in the Suprime-Cam 2 Square Degree Field*, Astrophys. J Letters, 580:L97-L100, 2002
- [12] T. Hamana *et al*, *Cosmic Shear Statistics in the Suprime-Cam 2.1 Square Degree Field: Constraints on Ω_m and σ_8* , Astrophys. J, 597:98-110, 2003
- [13] Y. Kamata *et al*, *Recent results of the fully-depleted back-illuminated CCD developed by Hamamatsu*, Proc. SPIE, Vol. 6276, 62761U, 2006
- [14] H. Furusawa *et al*, *Hyper Suprime-Cam: data analysis and management system*, Proc. SPIE, Vol. 7016, 70161F, 2008

- [15] H. Nakaya *et al*, *Hyper Suprime-Cam: CCD readout electronics*, Proc. SPIE, Vol. 7014, 70144X (2008)
- [16] Y. Komiyama, *et al*, *Hyper Suprime-Cam: camera dewar*, SPIE 7014, 70144V (2008)
- [17] IEEE, *Standard for Mechanical Core Specifications for Microcomputers Using IEC 60603-2 Connectors*, IEEE Standard 1101.1 (1998)
- [18] IEEE, *Standard for Additional Mechanical Specifications for Microcomputers Using the IEEE Std 1101.1-1991 Equipment Practice*, IEEE Standard 1101.1 (1996)
- [19] J.R.Janesick, *Scientific Charge-Couples Devices*, SPIE Press, Bellingham, p.578, 2001
- [20] Internet Engineering Task Force, *INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION*, RFC791, Sep, 1981
- [21] W. R. Stevens, *TCP/IP Illustrated, Volume1: The protocols*, Addison-Wesley, 1994
- [22] Internet Engineering Task Force, *Transmission Control Protocol*, RFC793, Sep, 1981
- [23] IEEE, *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE standard 802.3, 2003
- [24] T. Uchida *et al*, *Hyper Suprime-Cam: back-end electronics for CCD readout*, Proc. SPIE, Vol. 7014, 70146K (2008)
- [25] IEEE, *IEEE standard for a common mezzanine card (CMC) family*, IEEE Std 1386, 2001
- [26] T. Uchida, *Hardware-Based TCP Processor for Gigabit Ethernet*, IEEE Transactions on Nuclear Science, 55, 1631, 2008
- [27] H.Nakaya *et al*, *New Focal Plane Array Controller for the Instruments of the Subaru Telescope*, PASP, 118:478-488, 2006
- [28] Xilinx, *Virtex-5 FPGA User Guide*
- [29] JEDEC, *DDR2 SDRAM SPECIFICATION*, JEDEC Standard No. 79-2C, 2006
- [30] JEDEC, *DDR2 SDRAM 32b-SO-DIMM Reference Design Specification*, JEDEC Standard No. 21C 4.20.16-1, 2007
- [31] Tektronics, *TDSJIT3 Data Sheet*
- [32] Engineering Department, Electronic Industries Association, *EIA Standard RS-485 Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems*, reprinted in Telebyte Technology "Data Communication Library" Greenlawn NY, 1985
- [33] Epson Toyocom, *XG-1000CA Data sheet*
- [34] Analog Devices, *AD7677 Data sheet*