

2026年度(令和8年)版

Ver. 2026-10-02a

Course number: CSC.T363



コンピュータアーキテクチャ Computer Architecture

1. コンピュータの構成と動作原理 Guidance and Fundamentals of Computer Design



www.arch.cs.titech.ac.jp/lecture/CA/
Tue 13:30-15:10, 15:25-17:05
Fri 13:30-15:10

吉瀬 謙二 情報工学系
Kenji Kise, Department of Computer Science
kise_at_comp.isct.ac.jp

お知らせ

- 講義および演習は、**西8号館 2階 E207 情報工学系計算機室**で実施します。
 - 情報工学系計算機室に集まってください。 <http://www.csc.titech.ac.jp/>
- 講義の最新情報は次のサポートページを確認してください。
 - www.arch.cs.titech.ac.jp/lecture/CA/

Computer Architecture (CSC.T363) Support Page, Dept. of Computer Science, Science Tokyo

News

Lectures and exercises will be held in **the computer room of the department of computer science** on the 2nd floor of Ookayama Campus West 8 Building.

- This page was refreshed for the new quarter. (2026-09-30)
- This page was refreshed for the new quarter. (2025-10-02)
- This page was refreshed for the new quarter. (2024-10-03)
- This page was refreshed for the new quarter. (2023-10-03)
- This page was refreshed for the new quarter. (2022-09-23)
- This page was refreshed for the new quarter. (2021-09-30)
- This page is open. (2018-09-27)

Final Exam

- **2026-11-27 (13:30-15:10) final exam** in the computer room of the department of computer science

Materials for Computer Architecture Exercise

- Exercise_1 2026-10-06 (15:25-17:05): Exercise(1) Using an FPGA board, UART, Input/Output
- Exercise_2 2026-10-13 (15:25-17:05):
- Exercise_3 2026-10-20 (15:25-17:05):
- Exercise_4 2026-10-27 (15:25-17:05):

【重要】情報工学系 計算機室のアカウント

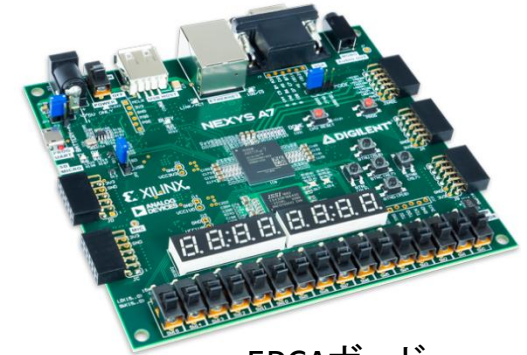
- 目の前のコンピュータにログインできない場合には、この講義の終了後にアカウントを作成する。
- コンピュータの **Ubuntu** 環境を利用します。



<https://www.csc.titech.ac.jp/>

この講義の特徴

- 講義2単位、演習1単位。
- 演習は、個人での作業
- 1人1台のFPGA (Field-Programmable Gate Array) ボードを用いた演習。
- 教科書で説明されるRISC-Vのコンピュータをハードウェア記述言語Verilog HDLで記述し、FPGAボードに実装する。
- コンピュータの高速化に取り組み、コンテスト形式で成果を競う。
- 質問などの情報共有には、講義専用の Slack を利用する。



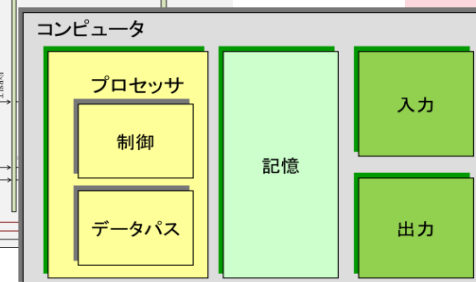
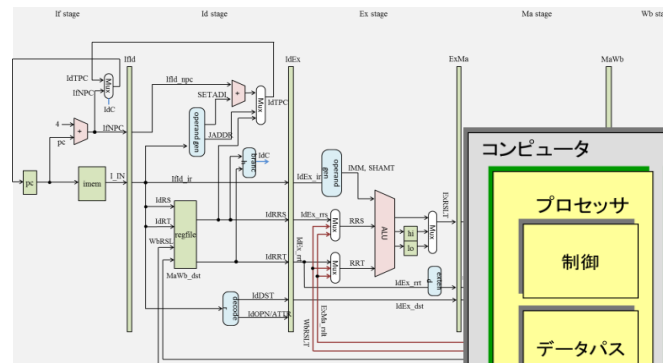
FPGAボード

```
module main (clk, led);
  input  wire clk;
  output wire led;

  reg [26:0] cnt;
  always @(posedge clk) cnt <= cnt + 1;

  assign led = cnt[26];
endmodule
```

Verilog HDL code



Digilent Nexys A7 FPGA board

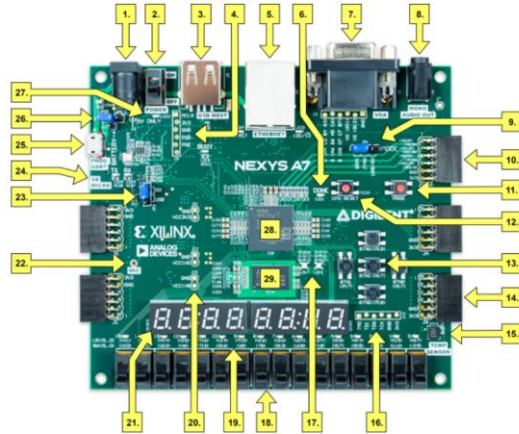


Figure 1. Nexys A7 Feature Callout

Callout	Component Description	Callout	Component Description
1	Power jack	16	JTAG port for (optional) external cable
2	Power switch	17	Tri-color (RGB) LEDs
3	USB host connector	18	Slide switches (16)
4	PIC24 programming port (factory use)	19	LEDs (16)
5	Ethernet connector	20	Power supply test point(s)
6	FPGA programming done LED	21	Eight digit 7-seg display
7	VGA connector	22	Microphone
8	Audio connector	23	External configuration jumper (SD / USB)
9	Programming mode jumper	24	MicroSD card slot
10	Analog signal Pmod port (XADC)	25	Shared UART/ JTAG USB port
11	FPGA configuration reset button	26	Power select jumper and battery header
12	CPU reset button (for soft cores)	27	Power-good LED
13	Five pushbuttons	28	AMD Artix™ 7 FPGA
14	Pmod port(s)	29	DDR2 memory
15	Temperature sensor		

Features

- **AMD Artix™ 7 FPGA**
 - 15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops (*8,150 slices)
 - 4,860 Kbits of fast block RAM (*2,700 Kbits)
 - Six clock management tiles, each with phase-locked loop (PLL)
 - 240 DSP slices (*120 DSPs)
 - Internal clock speeds exceeding 450 MHz
 - Dual-channel, 1 MSPS internal analog-digital converter (XADC)
- **Memory**
 - 128MiB DDR2
 - Serial Flash
 - microSD card slot
- **Power**
 - Powered from USB or any 4.5V-5.5V external power source
- **USB and Ethernet**
 - 10/100 Ethernet PHY
 - USB-JTAG programming circuitry
 - USB-UART bridge
 - USB HID Host for mice, keyboards and memory sticks
- **Simple User Input/Output**
 - 16 Switches
 - 16 LEDs
 - Two RGB LEDs
 - Two 4-digit 7-segment displays
- **Audio and Video**
 - 12-bit VGA output
 - PWM audio output
 - PDM microphone
- **Additional Sensors**
 - 3-axis accelerometer
 - Temperature sensor
- **Expansion Connectors**
 - Pmod connector for XADC signals
 - Four Pmod connectors providing 32 total FPGA I/O

Syllabus (1/3)

授業の目的（ねらい）、概要

情報工学分野において、コンピュータの動作原理と高性能化のための方式を理解し、さらにハードウェアとソフトウェアの相互関係を習得することは非常に重要です。本講義では、プロセッサ、メモリ、入出力、マルチプロセッサ、マルチコアというコンピュータシステムを構成する各種装置について、その役割と動作原理を学びます。

演習では、Verilog HDL等のハードウェア記述言語を用いてFPGAが搭載されたハードウェアボード等に高度なデジタル回路であるプロセッサを実装し、プログラムを動作させることでハードウェアとソフトウェアの相互関係を習得します。

到達目標

本講義を履修することによってコンピュータ以下を習得する。

- ・コンピュータの動作原理と高性能化のためのアーキテクチャ
- ・プロセッサ、メモリ、入出力、マルチプロセッサ、マルチコアといったコンピュータシステムを構成する各種装置の役割と動作原理
- ・ハードウェア記述言語を用いた一般的なコンピュータシステムの設計能力

キーワード

コンピュータアーキテクチャ、プロセッサ、スーパースカラ、キャッシュ、メモリ階層、仮想記憶、マルチプロセッサ、マルチコア、ハードウェア記述言語、Verilog HDL、FPGA

学生が身につける力

☒ 専門力 ☐ 教養力 ☐ コミュニケーション力 ☐ 展開力 (探究力又は設定力) ☒ 展開力 (実践力又は解決力)

授業の進め方

原則として、90分×2コマの講義の後、90分×1コマのFPGAボードを用いた演習をおこないます。

100分

<https://syllabus.s.isct.ac.jp/courses/2025/4/0-904-342300-0-0/202102443>

Syllabus (2/3)

授業計画・課題

	授業計画	課題
第1回	コンピュータの構成と動作原理	コンピュータの構成と動作原理について理解する。
第2回	コンピュータの性能と消費電力の動向	コンピュータの性能と消費電力の動向について理解する。
第3回	コンピュータ設計演習(1)	コンピュータ設計演習(1)
第4回	半導体メモリ	半導体メモリについて理解する。
第5回	コンピュータ設計演習(2)	コンピュータ設計演習(2)
第6回	キャッシュ：ダイレクトマップ方式	ダイレクトマップ方式のキャッシュについて理解する。
第7回	キャッシュ：セットアソシアティブ方式	セットアソシアティブ方式のキャッシュについて理解する。
第8回	コンピュータ設計演習(3)	コンピュータ設計演習(3)
第9回	メモリシステムの階層化と信頼性	メモリシステムの階層化と信頼性について理解する。
第10回	パイプラインプロセッサ	パイプラインプロセッサについて理解する。
第11回	コンピュータ設計演習(4)	コンピュータ設計演習(4)
第12回	スーパースカラプロセッサ	スーパースカラプロセッサについて理解する。
第13回	分岐予測	分岐予測について理解する。
第14回	コンピュータ設計演習(5)	コンピュータ設計演習(5)
第15回	ベクタ、SIMDにおけるデータレベル並列性	ベクタ、SIMDにおけるデータレベル並列性について理解する。
第16回	仮想記憶、セキュリティ	仮想記憶、セキュリティについて理解する。
第17回	コンピュータ設計演習(6)	コンピュータ設計演習(6)
第18回	入出力、バス	入出力、バスについて理解する。
第19回	相互接続ネットワーク	相互接続ネットワークについて理解する。
第20回	コンピュータ設計演習(7)	コンピュータ設計演習(7)
第21回	マルチプロセッサ、マルチコア	マルチプロセッサ、マルチコアについて理解する。

最新の情報はサポートページに掲載する。

Syllabus (3/3)

教科書

デイビッド・A. パターソン、ジョン・L. ヘネシー (著)、成田光彰 (翻訳) 『コンピュータの構成と設計 第5版 上/下』 日経BP社

参考書、講義資料等

無し。

成績評価の方法及び基準

講義で扱うコンピュータアーキテクチャに関する理解、ハードウェア記述言語を用いたコンピュータシステム実装への応用力を評価する。
演習 (30%) と期末試験 (70%) により評価する。

40%

60%

関連する科目

- CSC.T252 : 論理回路理論
- CSC.T262 : アセンブリ言語
- CSC.T372 : コンパイラ構成
- CSC.T341 : コンピュータ論理設計
- CSC.T433 : 先端コンピュータアーキテクチャ



講義日程(予定)

Final Exam

- **2026-11-27 (13:30-15:10) final exam** in the computer room of the department of computer science

Materials for Computer Architecture Exercise

- Exercise_1 2026-10-06 (15:25-17:05): Exercise(1) Using an FPGA board, UART, Input/Output
- Exercise_2 2026-10-13 (15:25-17:05):
- Exercise_3 2026-10-20 (15:25-17:05):
- Exercise_4 2026-10-27 (15:25-17:05):
- Exercise_5 2026-11-10 (15:25-17:05):
- Exercise_6 2026-11-17 (15:25-17:05):
- Exercise_7 2026-11-24 (**13:30**-17:05):

Lecture Slides and Materials

- Lecture_01 2026-10-02 (13:30-15:10): Guidance and Fundamentals of Computer Design
- Lecture_02 2026-10-06 (13:30-15:10): Trends in Performance and Power
- Lecture_03 2026-10-13 (13:30-15:10): Memory Technologies
- Lecture_04 2026-10-16 (13:30-15:10): Caches: Direct-Mapped
- Lecture_05 2026-10-20 (13:30-15:10): Caches: Set-Associative
- Lecture_06 2026-10-23 (13:30-15:10): CFU Proving Ground, Branch Prediction
- Lecture_07 2026-10-27 (13:30-15:10): Branch Prediction (2)
- Lecture_08 2026-10-30 (13:30-15:10): Pipelined Processor (1)
- Lecture_09 2026-11-06 (13:30-15:10): Pipelined Processor (2)
- Lecture_10 2026-11-10 (13:30-15:10): Virtual Memory and Dependability (1)
- Lecture_11 2026-11-13 (13:30-15:10): Virtual Memory and Dependability (2)
- Lecture_12 2026-11-17 (13:30-15:10): Superscalar, Data-Level Parallelism in Vector and SIMD, Input/Output and Bus
- Lecture_13 2026-11-20 (13:30-15:10): Interconnection Network, Multiprocessors and Multicore



教科書

コンピュータの構成と設計 第5版、パターソン&ヘネシー
(成田光彰 訳)、日経BP社



参考書

RISC-Vで学ぶコンピュータアーキテクチャ 完全入門
吉瀬謙二、技術評論社



アーキテクチャ(建築), Architecture



パルテノン神殿



世界最大のクフ王のピラミッド
1個約2.5tのブロックを 230～250万 個
積み重ねて造られている。

写真は計算機アーキテクチャのホームページから <http://www.cs.wisc.edu/arch/www/>





What's Computer Architecture?

Computer Architecture is the science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals. Computer architecture is *not* about using computers to design buildings.

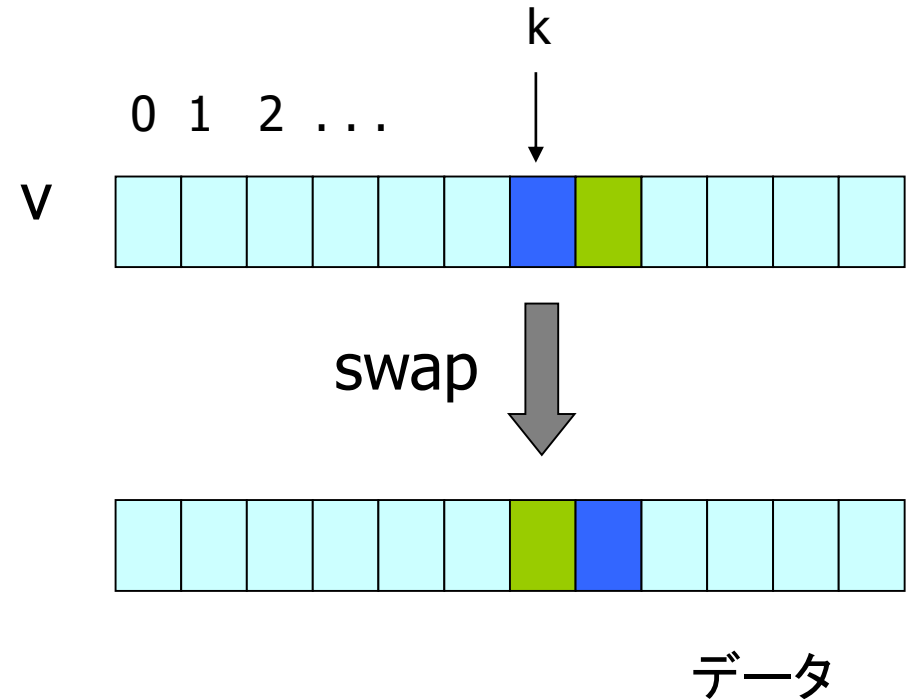
計算機アーキテクチャのホームページから <http://www.cs.wisc.edu/arch/www/>



高水準言語からハードウェアの言語へ

```
swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

C言語で記述したプログラム



RISC-V Instruction Set Architecture (ISA)



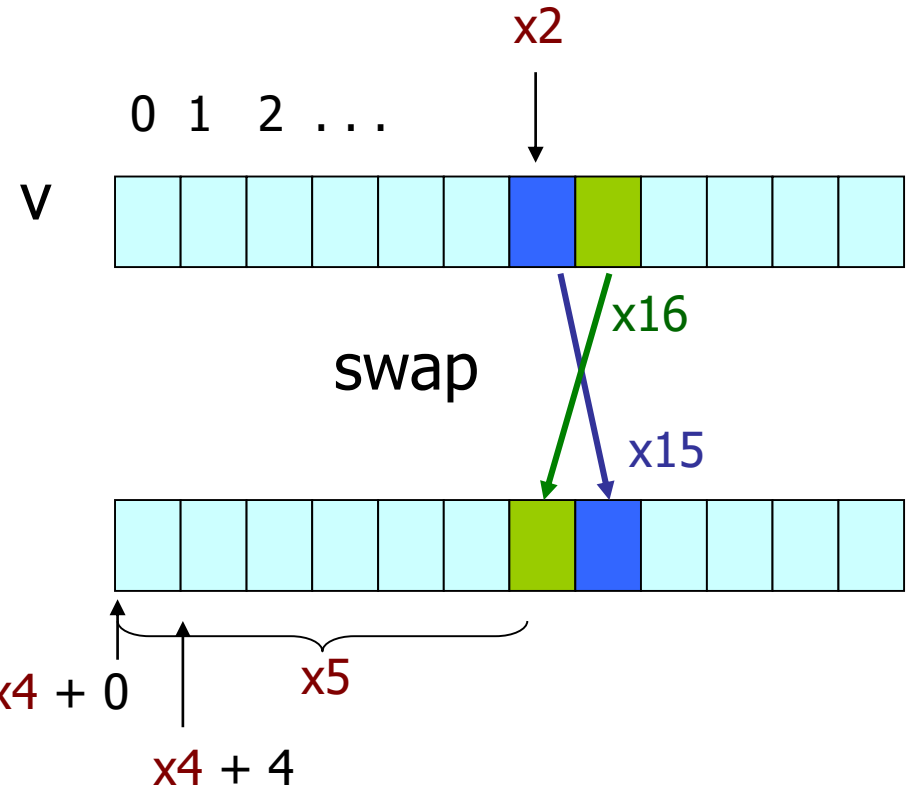
funct7	rs2	rs1	funct3	rd	opcode	R-type
0000000	rs2	rs1	000	rd	0110011	R-type add
0100000	rs2	rs1	000	rd	0110011	R-type sub
0000000	rs2	rs1	001	rd	0110011	R-type sll
0000000	rs2	rs1	010	rd	0110011	R-type slt
0000000	rs2	rs1	011	rd	0110011	R-type sltu
0000000	rs2	rs1	100	rd	0110011	R-type xor
0000000	rs2	rs1	101	rd	0110011	R-type srl
0100000	rs2	rs1	101	rd	0110011	R-type sra
0000000	rs2	rs1	110	rd	0110011	R-type or
0000000	rs2	rs1	111	rd	0110011	R-type and



高水準言語からハードウェアの言語へ

swap:

```
add  x2, x5, x5
add  x2, x2, x2
add  x2, x4, x2
lw   x15, 0(x2)
lw   x16, 4(x2)
sw   x16, 0(x2)
sw   x15, 4(x2)
ret
```



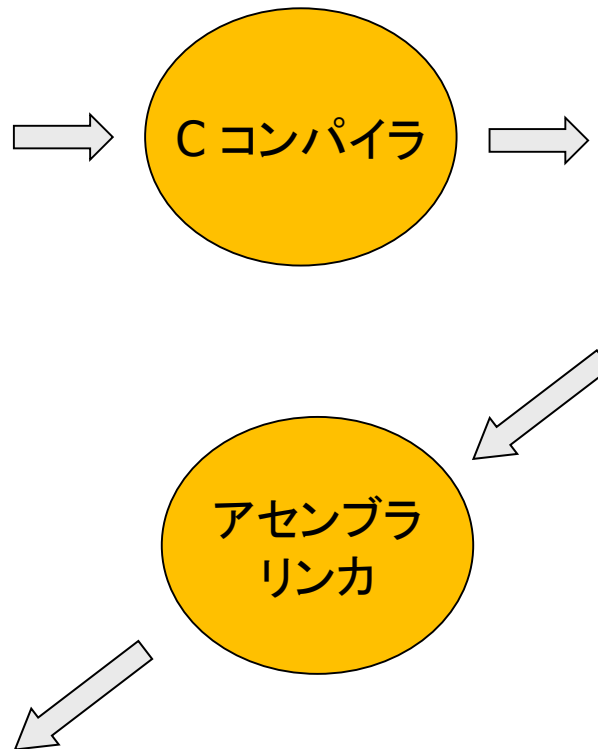
RISC-Vのアセンブリ言語に変換されたプログラム



高水準言語からハードウェアの言語へ

```
swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

C言語で記述したプログラム



swap:

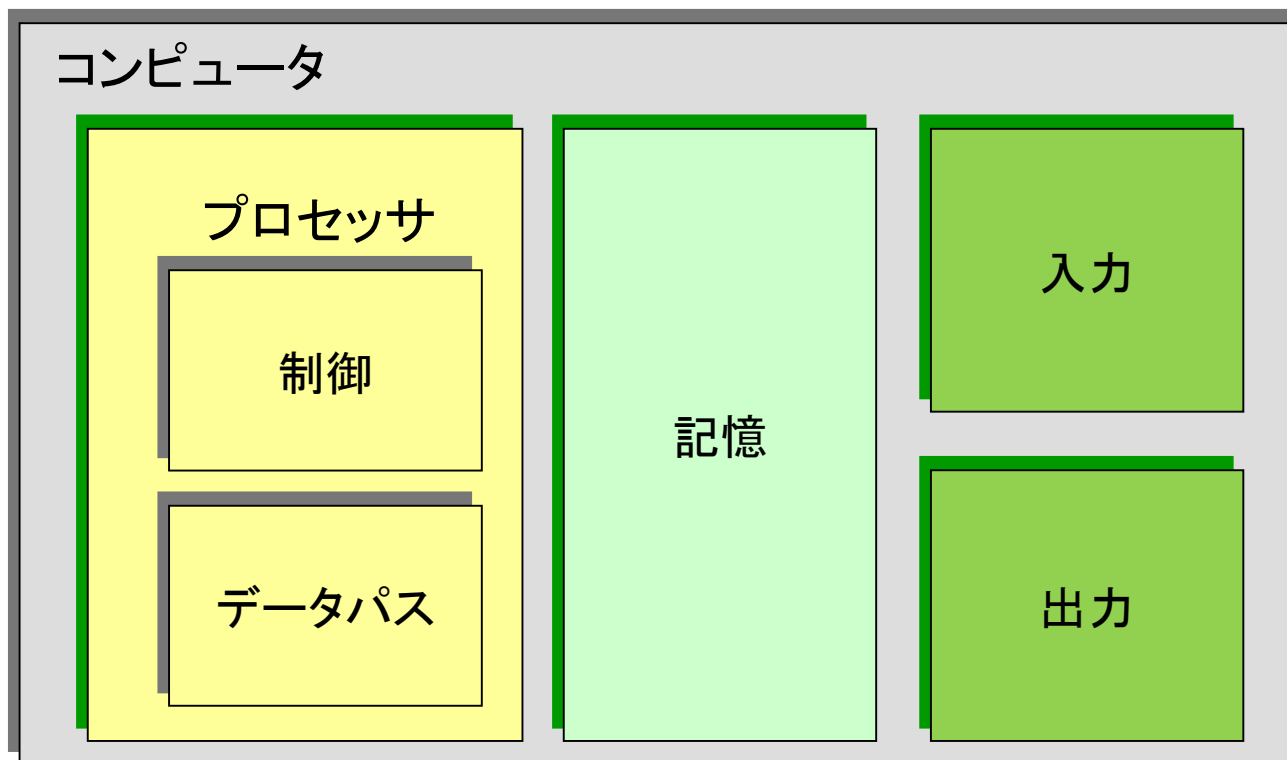
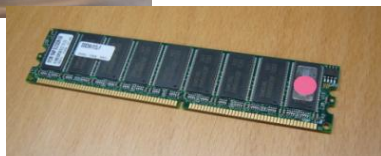
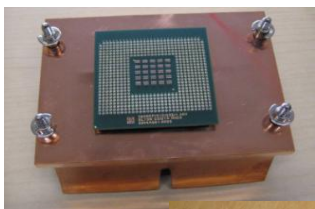
```
add    x2, x5, x5
add    x2, x2, x2
add    x2, x4, x2
lw     x15, 0(x2)
lw     x16, 4(x2)
sw     x16, 0(x2)
sw     x15, 4(x2)
ret
```

アセンブリ言語に変換されたプログラム

機械語に落とされたプログラム(機械命令の集まり)



コンピュータ(ハードウェア)の古典的な要素



プロセッサが実行する小さい単位の処理である命令(machine instruction)を記憶装置から取り出して、その指示に従ってプロセッサに格納されているデータ、あるいは記憶装置から取り出したデータに対する演算をおこなう。また、得られた演算の結果はプロセッサの内部に格納したり、記憶装置に格納する。コンピュータの外部からのデータを記憶装置に書き込むのが入力装置であり、記憶装置から読み出したデータを出力するのが出力装置である。記憶装置をメインメモリ、プロセッサをCPUと呼ぶことがある。

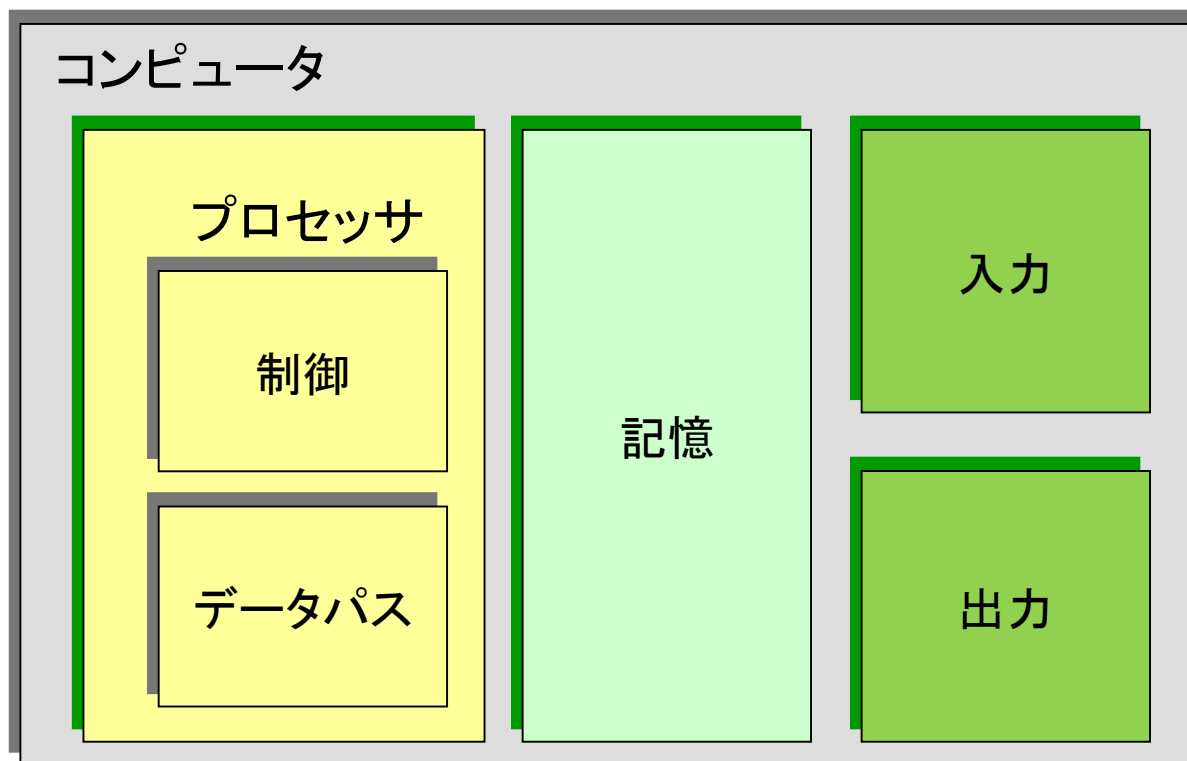
コンピュータの古典的な要素

コンパイラ

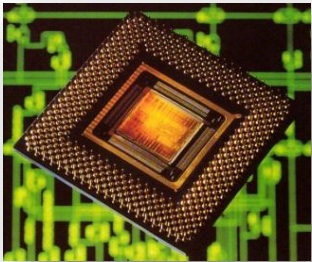
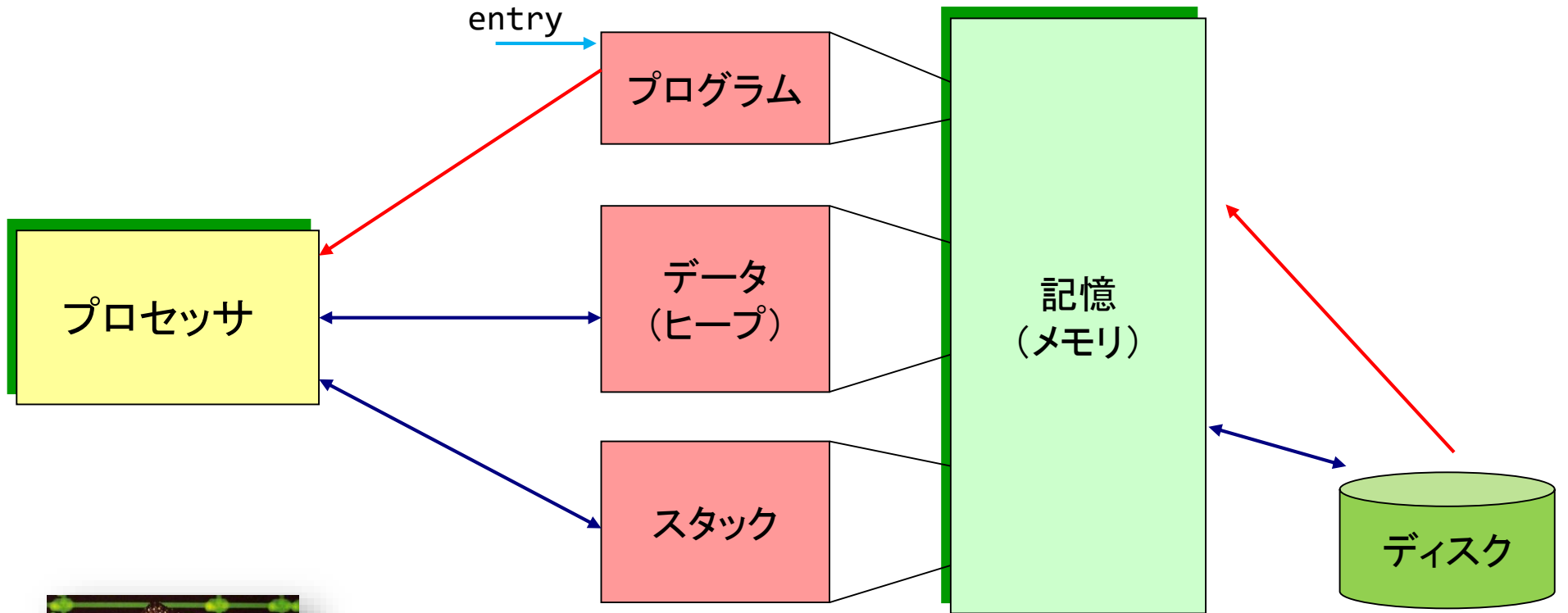
Instruction Set Architecture (ISA), 命令セットアーキテクチャ

インタフェース

性能の評価



プログラム, データ, その他



4GB (32bit) memory space and virtual memory

0x00000000

00000000 00000000 00000000 00000000₂ = 0₁₀



2GB Memory !

```
kterm
top - 11:35:26 up 10 days, 19:49, 2 users, load average: 0.01, 0.01, 0
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si,
Mem: 4002924k total, 3252404k used, 750520k free, 181808k buffers
Swap: 6062072k total, 0k used, 6062072k free, 2570804k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        15   0 10348   896   584  S   0.0   0.0   0:01.00  init
    2 root         RT  -5    0     0     0  S   0.0   0.0   0:00.00  migration/0
    3 root        34  19     0     0     0  S   0.0   0.0   0:00.00  ksoftirqd/0
    4 root         RT  -5     0     0     0  S   0.0   0.0   0:00.00  watchdog/0
    5 root         RT  -5     0     0     0  S   0.0   0.0   0:00.00  migration/1
    6 root        34  19     0     0     0  S   0.0   0.0   0:00.00  ksoftirqd/1
    7 root         RT  -5     0     0     0  S   0.0   0.0   0:00.00  watchdog/1
    8 root         RT  -5     0     0     0  S   0.0   0.0   0:00.01  migration/2
    9 root        34  19     0     0     0  S   0.0   0.0   0:00.00  ksoftirqd/2
   10 root         RT  -5     0     0     0  S   0.0   0.0   0:00.00  watchdog/2
```

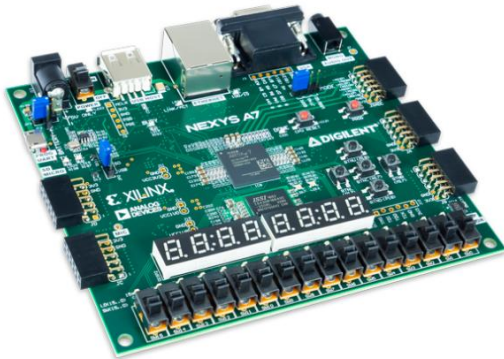
0xFFFFFFFF

11111111 11111111 11111111 11111111₂ = 4,294,967,296 - 1₁₀



FPGAボードのLEDを点滅させるデモ

- LEDの点滅
- Nexys A7 FPGA board
 - xc7a100tcs9324-1



```
module main (clk, led);
    input  wire clk;
    output wire [3:0] led;

    reg [31:0] cnt = 0;
    always @(posedge clk) cnt <= cnt + 1;

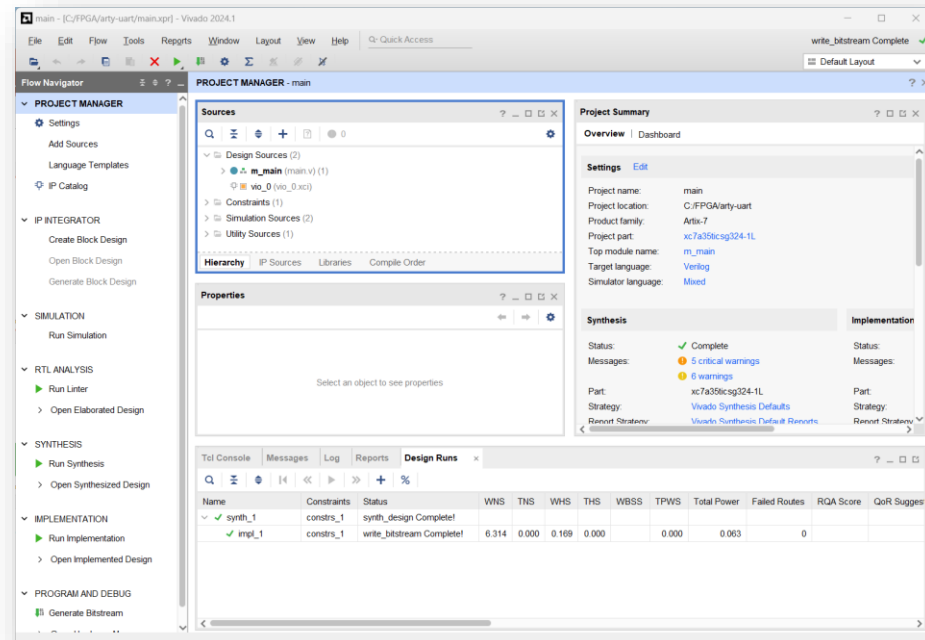
    assign led = cnt[27:24];
endmodule
```

main.v

```
set_property -dict { PACKAGE_PIN E3  IOSTANDARD LVCMOS33} [get_ports { clk }];
create_clock -add -name sys_clk -period 10.00 [get_ports { clk }];

set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33} [get_ports { led[0] }];
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33} [get_ports { led[1] }];
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33} [get_ports { led[2] }];
set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33} [get_ports { led[3] }];
```

main.xdc



<https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual>