８bit CPU の製作

荒瀬　統真

１．研究概要

コンピュータやプログラミングを学ぶ上でブラックボックス化しがちな CPU の基本的な仕組みについて理解するために、８bit の CPU を実際に製作した。

２．研究の具体的内容

（１）CPU について

CPU とはコンピュータの頭脳のようなもので、計算やコンピュータの各装置を制御するといった役割がある。今回私が製作した CPU を構成する主な装置は以下の通りである。

・レジスタ
　データを一時的に記憶しておく装置

・プログラムレジスタ
　メモリの番地を指定するための装置

・データセレクタ
　制御信号に応じてデータを選択する装置

・ALU（算術論理演算装置）
　制御信号に応じた演算をする装置

・メモリ
　プログラムを保存しておくための装置

・命令デコーダ
　メモリから送られたオペレーションコードを解読して、各装置に制御信号を送る装置

（２）設計について

今回製作する CPU は２年生の時に製作した４bit の CPU を改良したものである。８bit にすることで以下のような利点がある。

・データバスが８bit なので一度に８bit のデータを処理することができる

・メモリのアドレス幅が広がり、最大 256 種類のメモリ空間を使うことができる

・最大 255 までの数値を扱うことができる。（４bit の場合は 15）

またメモリには SRAM を使う､データセレクタを二つにするといった装置の変更のほかに、配線の手間を減らすためにワンボードではなく複数のモジュールを組み合わせるという構成にした。以下が今回製作した CPU のブロック図である。

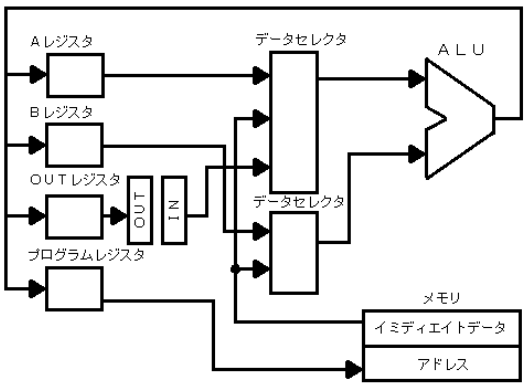

図１　CPU のブロック図

（３）モジュール基板

各装置のモジュール基板について説明する。

・レジスタ基板

この基板は図１で示した A レジスタ、B レジスタ､OUT レジスタ（データ出力用）といった３つのレジスタが載った基板

である。この３つのレジスタは 74HC161 で構成している。74HC161 自体は４bit のカウンタ IC だが、カウント機能をとれば LOAD ピンでデータの保持、ロードの制御を楽に行うことができる。また、同じ IC を二つ組み合わせて８bit のデータを扱えるようにしている。図２が製作したレジスタ基板である。

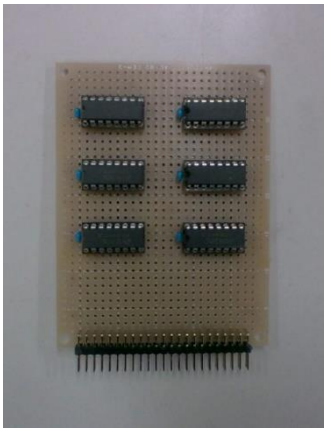

図２　レジスタ基板

・データセレクタ基板

この基板は図１で示した A レジスタ、入力ポート、イミディエイトデータの選択を行うデータセレクタが載った基板である。データセレクタは 74HC153 で構成している。４つ使うことで８bit のデータを２bit の制御信号に応じてデータを三種類から一つ選択できるようになっている。（図３）

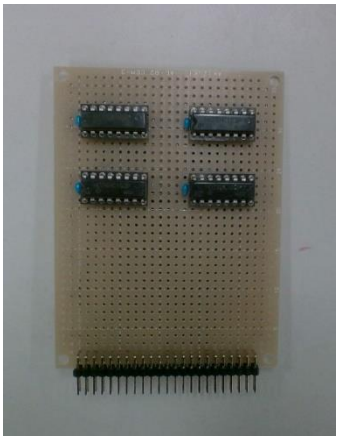

図３　データセレクタ基板

・プログラムレジスタ-データセレクタ基板

この基板は図１で示したプログラムレジスタ、データセレクタが載った基板である。プレグラムレジスタは 74H161 を使用している。また、データセレクタは図１で示した B レジスタとイミディエイトデータの選択を行うものである。図４が製作した基板である。

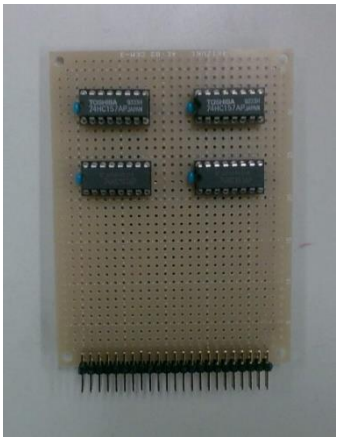

図４　データセレクタ基板

・ALU 基板

この基板は ALU（算術論理演算装置）が載った基板である。ALU は MC14581 を使用している。ALU は４bit の制御信号に応じて算術演算、論理演算を行うことができる。（図５）

SELECTION				ACTIVE-LOW DATA		
				M = H	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0	LOGIC FUNCTIONS	Cn = L (no carry)	Cn = H (with carry)
L	L	L	L	F = $\overline{A}$	F = A MINUS 1	F = A
L	L	L	H	F = $\overline{AB}$	F = AB MINUS 1	F = AB
L	L	H	L	F = $\overline{A}$ + B	F = A$\overline{B}$ MINUS 1	F = A$\overline{B}$
L	L	H	H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L	H	L	L	F = A + $\overline{B}$	F = A PLUS (A + $\overline{B}$)	F = A PLUS (A + $\overline{B}$) PLUS 1
L	H	L	H	F = $\overline{B}$	F = AB PLUS (A + $\overline{B}$)	F = AB PLUS (A + $\overline{B}$) PLUS 1
L	H	H	L	F = A ⊕ B	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = A + $\overline{B}$	F = A + $\overline{B}$	F = (A + $\overline{B}$) PLUS 1
H	L	L	L	F = $\overline{AB}$	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H	L	L	H	F = A ⊕ B	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = A$\overline{B}$ PLUS (A + B)	F = A$\overline{B}$ PLUS (A + B) PLUS 1
H	L	H	H	F = A + B	F = (A + B)	F = (A + B) PLUS 1
H	H	L	L	F = 0	F = A PLUS A^2	F = A PLUS A PLUS 1
H	H	L	H	F = A$\overline{B}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H	H	H	L	F = AB	F = A$\overline{B}$ PLUS A	F = A$\overline{B}$ PLUS A PLUS 1
H	H	H	H	F = A	F = A	F = A PLUS 1

図５　真理値表（データシートより抜粋）

IC を二つ組み合わせることで８bit の演算を可能にしている。また D フリップフロップの 74HC74 を使いキャリーを保持できるようにしている（キャリーはキャリーフラグ

として条件分岐に使用)。図６が製作した基板である。

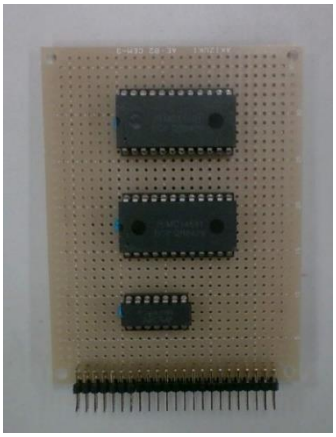

図６　ALU 基板

・メモリ基板

この基板はメモリ IC とメモリのバッテリーバックアップ回路が載った基板である。メモリには SRAM の LH5116 を使用している。二つ組み合わることでアドレス幅を８bit に増やすことで使用できるメモリ空間を 256 種類に増やしている。また、揮発性のメモリである SRAM を使用しているので､ボタン電池によるバッテリーバックアップ機能を搭載することで本体の電源を切ってもバックアップ電源に切り替わり、内部のデータを保持できるようにしている。図７が製作した基板である。

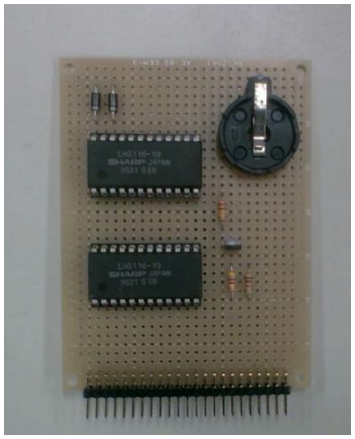

図７　メモリ基板

・命令デコーダ基板

本来、命令デコーダは論理回路で構成するものだが、８bit に拡張したことでデコーダ回路が煩雑になるため UV-EPROM を使っている。UV-EPROM は中心にある窓に紫外線を当てると内部データが消える揮発性のメモリである。データを書き込むために別途ライターを自作した。図８が製作した基板である。

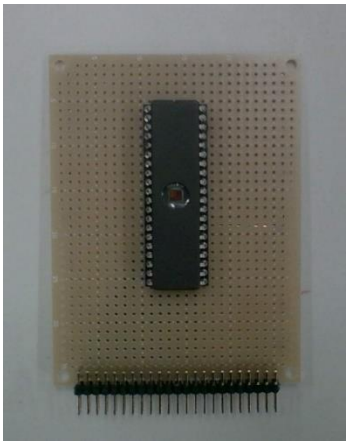

図８　命令デコーダ基板

（３）周辺装置

CPU が動作するためにはクロックジェネレータやパワーオンリセット回路、プログラムを書き込むためのライターといった周辺装置が必要である。以下に今回製作した CPU の周辺装置を紹介する。

・クロック回路

そもそも CPU という装置はクロック信号を送ることで動作を繰り返し、処理を行う装置である。クロック周波数（１秒間あたりのクロック回数）が高ければ高いほど高速に処理を行うことができるが、内部のトランジスタの動作速度、発熱、配線遅延、材料の特性などによって最大クロック周波数は変わってくる。クロックジェネレータには水晶発振器を用いることが一般的だが、今回製作する CPU では２年生の実習で学習した非安定

マルチバイブレータ回路を用いる。クロック周波数は１Hz と可変（３Hz～３kHz）を切り替えることができる。しかし抵抗とコンデンサの誤差や 74HC14 の特性によって実際の周波数は異なる。そのため１Hz の部分には可変抵抗を加えて微調整を可能にしている。また手動クロックも備えておりチャタリング防止のために CR フィルタとシュミットトリガを使用している。(図９、10)

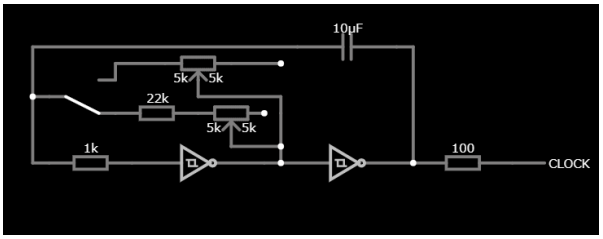

図９　クロック回路

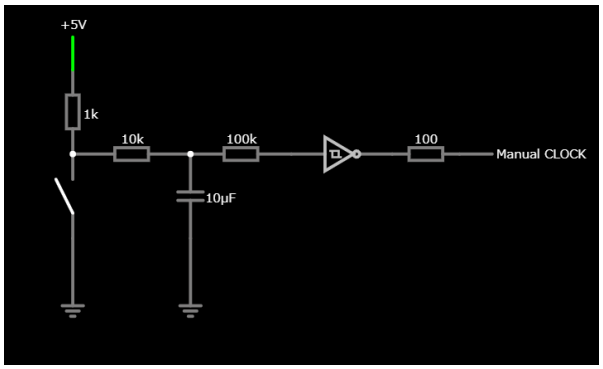

図 10　手動クロック回路

・パワーオンリセット回路

CPU におけるリセット回路とは､CPU を正しく動作させるための初期状態を作り出す回路である。電源投入時や異常発生時に CPU の状態を強制的に初期化して動作を再開できるようにしている。今回製作したリセット回路には放電用のスイッチがついており、任意のタイミングでリセット動作を行うことができる。図 11 がパワーオンリセット回路の回路図である。

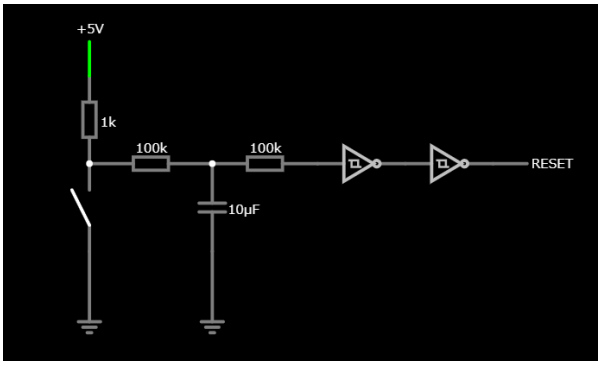

図 11　リセット回路

・ROM ライター

命令デコーダを UV-EPROM を使って構成しているため、データ書き込み用の ROM ライターが別途必要である。しかし市販品は値が張るため、自作した。(図 12)

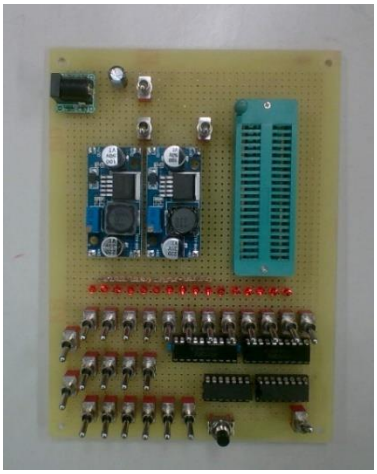

図 12　ROM ライター

トグルスイッチでアドレスと書き込むデータを指定し、プッシュスイッチを押すとつながっているワンショット回路から約 100μｓの LOW パルスが ROM のチップイネーブルに入力され、データが書き込まれる。ワンショット回路には単安定マルチバイブレータ IC の 74HC123 を使用している。また ROM にデータを書き込む際には VCC に 6.25V､VPP に 12.75V の電圧を印加する必要がある。必要な電圧を用意するために、市販の DC/DC コンバータを使用している。またこの ROM ライターには書き込みのほかに ROM 内のデータを LED への出

力で確認できるようにしており、トグルスイッチで書き込み、確認モードを切り替えることができる。

・SRAM ライター

図７のメモリ基板にプログラムを書き込むための装置である。回路自体は ROM ライターとほぼ同じだが、SRAM は書き込み時に電源電圧を上げる必要がないので DC/DC コンバータが無い分すっきりとしている。また、モジュール基板を差し込んでそのままプログラムを書き込むことができる。図 13 が製作した装置である。

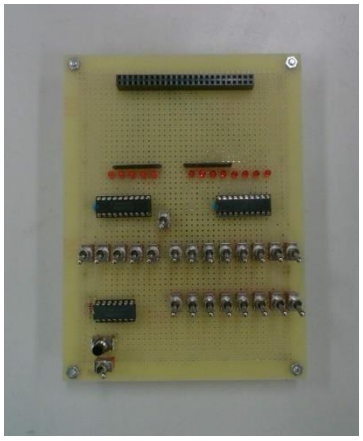

図 13　RAM ライター基板

・マザーボード

製作したモジュール基板をこのマザーボードにすべて差し込むことで CPU として動作する。このマザーボードにはクロック回路、リセット回路、入出力ポートが載っている。また、レジスタ内部のデータの様子を確認するための LED 基板（図 14）を差し込むスロットも用意している。

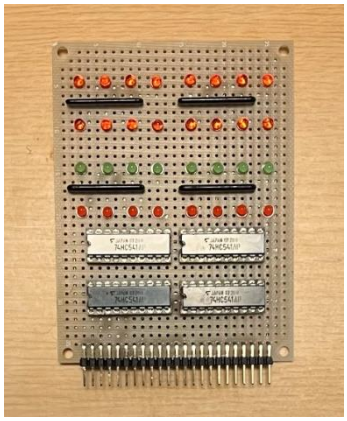

図１４　LED 基板

データ確認用の LED は上から A レジスタ、B レジスタ、OUT レジスタ、プログラムレジスタのものである。また入出力ポートは左下のピンソケットである。

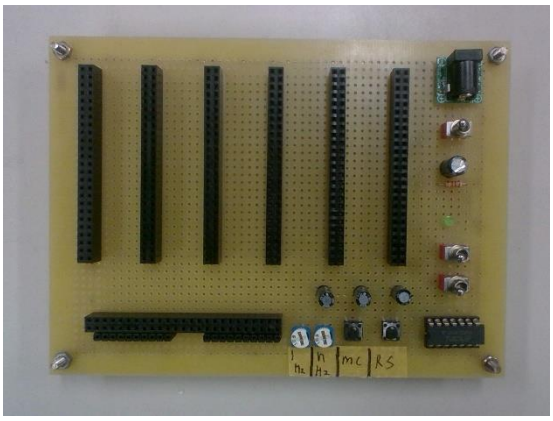

図 15　マザーボード

（４）命令について

命令とは CPU が実行する基本的な動作や処理内容を記述したものである。命令は機械語で書かれており、CPU が直接解釈して実行する。今回製作した CPU の命令の構成は上位５bit がオペレーションコード、下位８bit がイミディエイトデータの計 13bit である。以下に今回製作した CPU の命令を示す。

表１　算術演算命令

命令	説明
ADD A,Im	A レジスタに Im を加算する。
ADD B,Im	B レジスタに Im を加算する
ADD A,B	A レジスタと B レジスタの内容を加算し、A に格納する。
SUB A,Im	A レジスタから Im を減算する。
SUB B,Im	B レジスタから Im を減算する。
SUB A,B	A レジスタの内容から B レジスタの内容を減算する。

表２　論理演算命令

命令	説明
NOT A	A レジスタの内容を論理反転する。
NOT B	B レジスタの内容を論理反転する。
AND A,Im	A レジスタと Im の論理積をとる。
AND B,Im	B レジスタと Im の論理積をとる。
AND A,B	A レジスタと B レジスタの論理積をとる。
OR A,Im	A レジスタと Im の論理和をとる。
OR B,Im	B レジスタと Im の論理和をとる。
OR A,B	A レジスタと B レジスタの論理和をとる。
XOR A,Im	A レジスタと Im の排他的論理和をとる。
XOR B,Im	B レジスタと Im の排他的論理和をとる。
XOR A,B	A レジスタと B レジスタの排他的論理和をとる。
XNOR A,Im	A レジスタと Im の否定排他的論理和をとる。
XNOR B,Im	B レジスタと Im の否定排他的論理和をとる。
XNOR A,B	A レジスタと B レジスタの否定排他的論理和をとる。

表３　データ転送命令

命令	説明
IN A	入力ポートから A レジスタにデータを読み込む。
IN B	入力ポートから B レジスタにデータを読み込む。
MOV A,B	B レジスタの内容を A レジスタに転送する。
MOV A,Im	Im を A レジスタに転送する。
MOV B,A	A レジスタの内容を B レジスタに転送する。
OUT B	B レジスタの内容を出力ポートに送る。
OUT Im	Im を出力ポートに送る。

表４　制御命令

命令	説明
NOP	何もしない（No Operation）。
JNC （C=0）	キャリーフラグが 0 の場合にジャンプする。
JNC （C=1）	キャリーフラグが 1 の場合にジャンプする。
JMP	無条件に指定されたアドレスにジャンプする。

（５）完成図

以下がマザーボードにモジュール基板をすべて差し込んだ完成図である。（図 16、17）

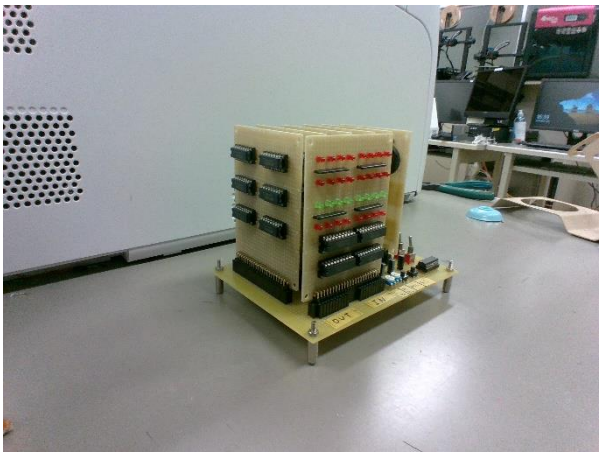

図 16　完成図１

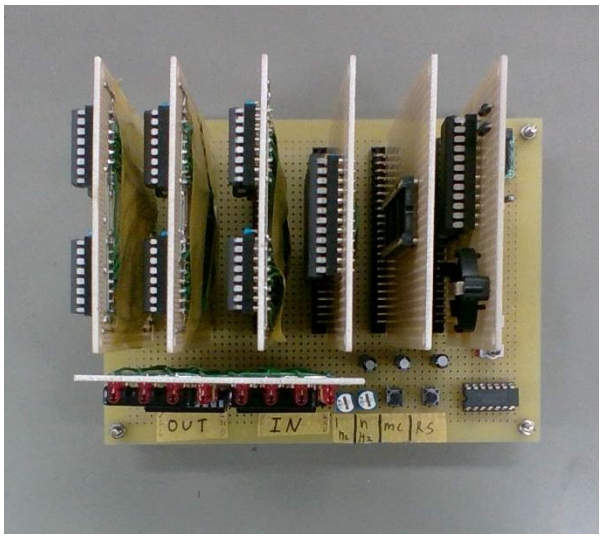

図 17　完成図２

（６）今後の展望

今回製作した８bit CPU の活用について以下のように考えている。

・周辺機器との接続による活用

今回製作した CPU には入出力ポートが備わっているため、センサーを接続して、センサーからのデータを CPU で処理し、モーターや LED を制御する簡単なシステムを構築することができる。また、LCD や７セグメント LED を接続し、CPU の演算結果や命令実行の様子を視覚的に表示することも可能である。

・FPGA を用いた再設計

FPGA（Field Programmable Gate Array）とは、HDL（ハードウェア記述言語）を用いて内部の回路設計を自由に書き換えられる IC のことである。回路設計を変更できるため、何度でも試行錯誤が可能で、レジスタや ALU、メモリの構成を変更することで、新しい機能を柔軟に追加することができる。FPGA を用いて同じ仕様の CPU を制作することで、FPGA と CMOS という実装方法の違いによる処理速度や消費電力を比較し、設計の最適化や実用性の向上につなげることができる。

３．研究のまとめ

今回の８bit CPU の製作を通して CPU の基本的な構造や仕組みについて学ぶことができた。レジスタやメモリなど実物を見る機会が少なく、プログラミングでイメージし辛い部分も実際に製作することで役割を理解することができた。また、モジュール基板を作ることによってワンボードに実装するよりも配線は少し楽になったが、それでも配線の量がかなり多く製作に時間がかかってしまった。CAD を用いてプリント基板を作ればもう少し製作期間を短くすることができたかもしれない。このような作業を通じて、ハードウェアの設計や組み立ての複雑さを実感した。一人での研究だったという点についてもやるべきことが多く、時間や労力をかなり費やしたがその分すべての工程を自分で行うことができ、自分の思う通りの製作ができたという達成感を得ることができた。

参考文献

CPU の創りかた
渡波　郁　マイナビブックス
https://book.mynavi.jp/ec/products/detail/id=22065#

74HC161（レジスタ）
https://toshiba.semicon-storage.com/info/TC74HC161AF_datasheet_ja_20140301.pdf?did=10806&prodName=TC74HC161AF

74HC153（データセレクタ）
https://toshiba.semicon-storage.com/info/TC74HC153AP_datasheet_ja_20140301.pdf?did=10307&prodName=TC74HC153AP

74HC157（データセレクタ）
https://toshiba.semicon-storage.com/info/TC74HC157AP_datasheet_ja_20140301.pdf?did=10663&prodName=TC74HC157AP

74HC74（D フリップフロップ）
https://toshiba.semicon-storage.com/info/TC74HC74AP_datasheet_ja_20140301.pdf?did=16716&prodName=TC74HC74AP

74HC00（NAND ゲート）
https://toshiba.semicon-storage.com/info/TC74HC00AP_datasheet_ja_20140301.pdf?did=6907&prodName=TC74HC00AP

74HC123（単安定マルチバイブレータ）
https://toshiba.semicon-storage.com/info/TC74HC123AF_datasheet_ja_20161202.pdf?did=8496&prodName=TC74HC123AF

74HC14（シュミットトリガインバータ）
https://toshiba.semicon-storage.com/info/TC74HC14AP_datasheet_ja_20140301.pdf?did=9945&prodName=TC74HC14AP

74HC541（３ステートバスバッファ）
https://toshiba.semicon-storage.com/info/TC74HC541AP_datasheet_ja_20140301.pdf?did=16392&prodName=TC74HC541AP

lh5116（SRAM）
https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/865/LH511610.pdf

m27c4002（UV-EPROM）
https://akizukidenshi.com/goodsaffix/m27c4002.pdf

mc14581（ALU）
https://mil.ufl.edu/4712/docs/MC14581B.pdf