第1章　コンピュータの基本構造

学習目標

コンピュータの発達の歴史と、コンピュータを構成している主要部分（入力装置、制御装置、演算装置、記憶装置、出力装置）の機能、相互の関連と役割について理解させる。

さらに、コンピュータ内部における情報の表現の仕方、基数、コード体系、数値の表現の仕方について習得させる。

内容のあらまし

内　容	説　明	議　論
コンピュータ発達の歴史	・コンピュータ以前の計算機械の発達を説明 ・コンピュータの発達と論理素子の発達の対応を説明 ・コンピュータの応用分野の広がりについて説明	コンピュータが生まれた背景普及を踏まえて今後どのような応用が考えられるかについて議論させる
コンピュータの基本構造	・コンピュータシステムがどのような装置で構成されているかを説明 ・各装置の名称、接続関係機能を説明 ・制御装置、演算装置、記憶装置、入力装置、出力装置が、それぞれどのような機能を担当し、接続された装置にどのような制御が行われ、入出力装置を通じて情報（データ）が受け渡されているかを説明	各装置は構成要素として何に対応するかを理解させる
基数	・数値の表現の方法が幾通りもある事を説明 ・基数が示されれば、その基数で表現された値を別の基数で示す事ができる事を説明	

内　　容	説　　明	議　　論
コード体系	・コンピュータ上で文字（英字、数字、カナ文字、特殊文字）を表すコードの体系を説明 ・漢字を表すためのコードについても説明	
数値の表現	・数値の２つの表現形式（固定小数点、浮動小数点）について、その表現方法、表現の違い、各々の利点を説明 ・２つの表現形式で表現可能な最大数、最小数の求め方を説明 ・１０進数の２つの表現形式（パック形式、ゾーン形式）について、その特徴を説明	

１．１　コンピュータ発達の歴史

１．１．１　コンピュータ以前の計算機械

現在コンピュータや電卓以外、通常目にする計算のための道具は、「そろばん」、「計算尺」などである。これらが持つ、計算に最低限必要な要素は以下の３つである。

① 表示している数字を人が意識的に変える事ができる（入力および制御ができる）。

② 数字が途中で人の意志に反して変化しない（数字を記憶できる）

③ 結果がわかる（出力ができる）

これを満足する計算機械は古くから存在しており、コンピュータの発展と大きくかかわってきた。コンピュータの発展過程上重要と思われるトピックは、以下の通り。

年　代	トピック
１７世紀初め	ネピア（J.Napier）が対数表を用いた「乗算機」を発明
１６４２年	パスカル（B.Pascal，フランス）が歯車を組み合わせた桁上がりができる「加算機」を発明
１６７１年	ライプニッツ（G.W.Leibnitz，ドイツ）が加算を繰り返し行うことで乗算を実行する「乗算機」を発明
１８３３年	バベジ（C.Babbage,イギリス）が、コンピュータの原型である「解析エンジン」を発明
１８８０年	ホレリス（H.Hollerith,アメリカ）が、パンチカードを発明、１９００年代にＰＣＳ（Punch Card System）へ発展する。

表１．１－１　コンピュータ発展過程上の主なトピック

１．１．２　コンピュータの発達

コンピュータの発達は、使用されている素子の内容により第１世代～第４世代と呼ばれている。この世代順に解説する。

①　第１世代（１９４５年～１９５０年代）

・論理素子　真空管
・主記憶素子　磁気ドラム、磁気コアメモリ、ブラウン管メモリ
・ハードウェアの特徴　固定小数点演算器
・ソフトウェアの特徴　機械語、アセンブリ言語
・代表的なコンピュータ

ＥＮＩＡＣ（１９４５年）：
ペンシルバニア大学のモークリ、エッカートなどにより開発され、最初の電子計算機とされている。
ＥＤＳＡＣ（１９４９年）：
ウイルクスなどにより開発。ストアドプログラム（stored Program）方式。
ＥＤＶＡＣ（１９５０年）：
フォン・ノイマンにより開発。ストアドプログラム方式。
ＵＮＩＶＡＣ　Ｉ（１９５１年）：
世界最初の商用計算機
ＩＢＭ７０１（１９５２年）
ＩＢＭ６５０（１９５３年）
ＩＢＭ７０４（１９５３年）
パラメトロン計算機（１９５４年）：
日本で開発された、論理素子としてパラメトロンを使用する計算機のちに　Ｍｕｓａｓｈｉｎｏ　Ｉとなる。

②　第２世代（１９５０年代後半～１９６０年代前半）

・論理素子　トランジスタ
・主記憶素子　磁気コアメモリ
・ハードウェアの特徴　浮動小数点演算器、インデックスレジスタ、Ｉ／Ｏプロセッサ
・ソフトウェアの特徴　高級言語、サブルーチンライブラリ、バッチモニタコンパイラ
・代表的なコンピュータ

ＩＢＭ７０７０，７０９０（１９５９年）
ＵＳＳＣ（１９５９年）
ＵＮＩＶＡＣⅢ（１９５９年）

③　第３世代（１９６０年代前半～１９６０年代いっぱい）

・論理素子　ＩＣ（集積回路）
・主記憶素子　磁気コアメモリ
・ハードウェアの特徴　マイクロプログラム、パイプライン処理、キャッシュメモリ、ミニコンピュータ
・ソフトウェアの特徴　マルチプログラミング、マルチプロセシング、オペレーティング・システム、仮想メモリ
・代表的なコンピュータ

ＩＢＭシステム／３６０（１９６４年）
ミニコンピュータの登場
日本電気ＮＥＡＣシリーズ
富士通２３０シリーズ
日立８０００シリーズ

④　第3.5世代（１９７０年代）

・論理素子　　ＬＳＩ（大規模集積回路）
・主記憶素子　　半導体メモリ
・ハードウェアの特徴　　多重パイプライン、内蔵アレイプロセッサ、スーパーコンピュータ、マルチプロセッサ
・ソフトウェアの特徴　　多重仮想メモリ、仮想マシン、データベース
・代表的なコンピュータ
　　ＩＢＭシステム／３７０（１９７０年）
　　マイクロコンピュータの登場
　　日本電気・東芝ＡＣＯＳシリーズ
　　富士通・日立Ｍシリーズ
　　沖・三菱ＣＯＳＭＯシリーズ

⑤　第4世代（１９８０年代以降）

・論理素子　　ＶＬＳＩ（超大規模集積回路）
・主記憶素子　　半導体メモリ
・ハードウェアの特徴　　コンピュータネットワーク、ネットワークサーバ、ワークステーション、パーソナルコンピュータ
・ソフトウェアの特徴　　分散処理、分散ＯＳ、エキスパートシステム
・代表的なコンピュータ
　　ＩＢＭシステム３０８１
　　ＩＢＭ４３３１、４３４１
　　スーパーミニコン
　　ワークステーション登場
　　パソコン登場
　　ＲＩＳＣ（Reduced Instruction Set Computer）

ＲＩＳＣマシンは、命令体系や命令セットを可能な限り簡素化したアーキテクチャで、頻繁に使用される命令を高速に実行することにより全体のパフォーマンスを上げる。従来複雑な命令で実現していた機能は、コンパイラがいくつかの簡単な命令の組み合わせにコンパイルする事により実現する。
ＲＩＳＣマシンに対して複雑な命令体系や豊富な命令セットを持つアーキテクチャをＣＩＳＣ（Complex Instruction Set Computer）マシンという。

１．１．３　コンピュータの応用分野

コンピュータは社会のあらゆる分野に浸透し、応用分野は多様化し、利用形態が複雑化している。応用分野は、大きく「事務計算」、「科学技術計算」、「制御」の３つに分ける事が出来るが、ファミリーコンピュータなどアミューズメント分野などや家庭電化製品にもコンピュータが内蔵されている。

事務分野では、バッチ処理からデータベース／データ通信へと、大容量化、ネットワーク化が押し進められている。また、基幹業務を支援するシステムのみならず、企業内の定型化できない情報を流通させるための情報系のシステムの構築が始まっている。

科学技術計算の分野では、高速演算、とりわけ浮動小数点演算の高速化が最大の課題であり、また大規模数値モデルによる解析には巨大なメモリ空間が必要となる。科学技術計算専用のスーパーコンピュータは、シミュレーションなどの大規模な数値計算の必要性が増大したために、証券、銀行など本来科学技術計算とは縁の薄い分野でも、大いに利用されるようになった。

制御分野においては、石油化学プラントや製鉄プラントなどの大規模生産設備の制御、交通輸送システムの制御等にみられるように、いったん事故が発生すると被害が大きくなる可能性が高いので、高度の信頼性が要求されている。

１．１．４　ハードウェアの構築形式

経済的な面や信頼性に対する要請からハードウェアのシステム構築形式にはいくつかの種類がある。

①　シンプレックスシステム（simplex system）

シンプレックスシステムは、中央処理装置、主記憶装置、入出力装置、外部記憶装置、通信制御装置などをそれぞれ１台ずつもつ最も簡単なシステムで、経済的だが本体が１つのために何らかの障害が発生するとシステムダウンの状態になりやすいシステムである。
信頼性よりも経済性を重視したシステムである。

図１．１－１　シンプレックスシステム

②　デュプレックスシステム（duplex system）

デュプレックスシステムは、待機システム（stand-by system）ともいい、システム全体の信頼性を上げるため通信制御装置、中央処理装置、主記憶装置、外部記憶装置、入出力装置などにたいしてそれぞれ予備機をもたせる方式である。
通常、一方の系がオンライン処理（優先度の高い処理・主系）を行い、他方の系がバッチ処理など（優先度の低い処理・従系）を行っており、主系に障害が発生した場合には従系の処理を中断して主系の処理の肩代わりをするという構成である。

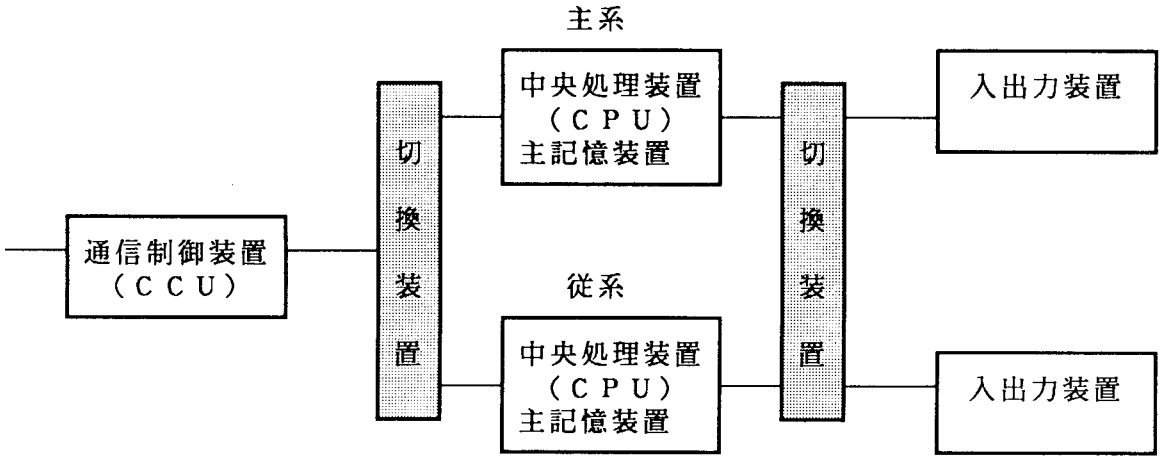


図１．１－２　デュプレックスシステム

③　デュアルシステム（dual system）

デュアルシステムは、デュプレックスシステムと同様に各装置を二重化しておき、完全な並列処理（同じ処理を二つのシステムで行う）を行い命令ごとに処理結果の照合チェックを行う（これをクロスチェックという）。結果が不一致のときは、直ちに診断処理を開始し、障害の発生した本体を切り離して処理を続ける。

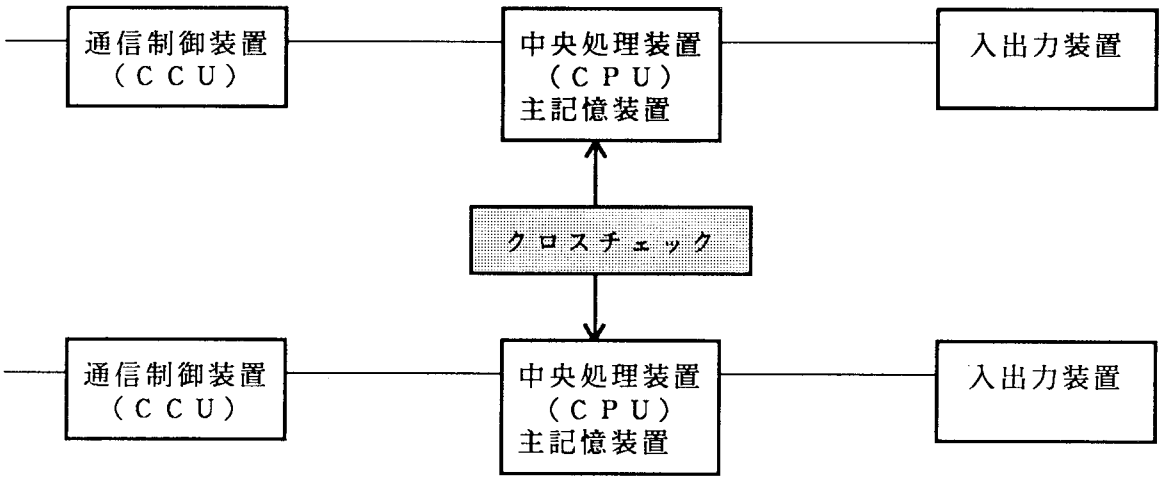


図１．１－３　デュアルスシステム

④　マルチプロセッサシステム（multiprocessor system）

マルチプロセッサシステムは、多重プロセッサシステムとも呼ばれ、複数の中央処理装置が主記憶装置を共用するシステムである。一つのオペレーティング・システムの下でシステム全体を制御し、いくつかの処理をそれぞれの中央処理装置が時分割方式で処理を行う。中央処理装置の間のかかわり合いを疎にしておくと中央処理装置に障害が発生した場合、障害の発生したものを切り離して処理を続行する事も可能である。また、本質的に同時に処理が行えるものを別々の中央処理装置に割り当てて、全体としての処理効率を上げる事も可能である。

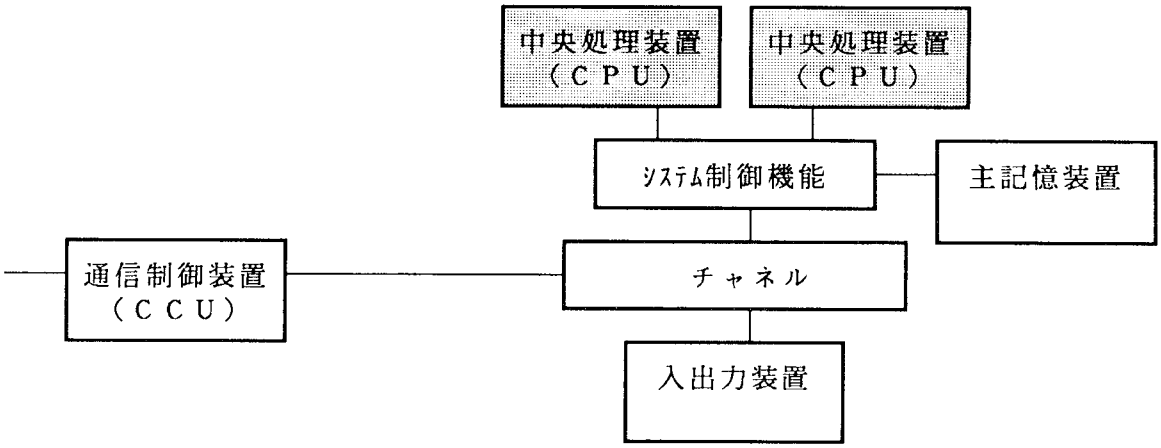


図１．１－４　マルチプロセッサシステム

⑤　タンデムシステム（tandem system）

伝送制御（メッセージ）などの前処理を行う計算機（Front-end processor）と適用業務を行う計算機と、データベースに対するアクセス等を行う計算機（Back-end processor）を直列に連結し、負荷の軽減と処理能力の向上を図ったシステムである。

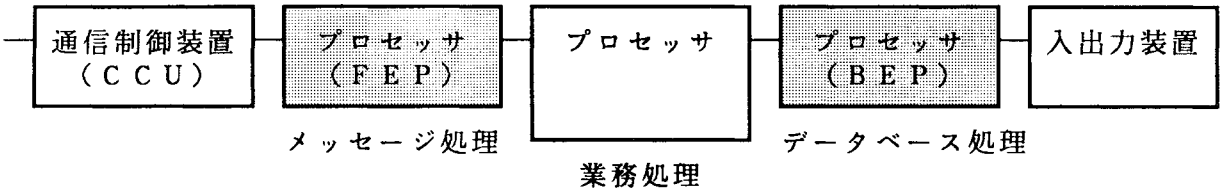


図１．１－５　タンデムセッサシステム

指導上の留意点

ポイント

① コンピュータの発明につながる重要事項について理解させる。計算方式の思想・原理を紹介し理解させる事も重要である。

② 論理素子の発達とコンピュータの発達の関係を、世代ごとに理解させる。各世代の代表的なコンピュータの特徴を理解させる。可能であれば、写真、ビデオなどを通じてコンピュータの成長過程を具体的なイメージで正しく認識させる。

③ 各世代ごとにハードウェアとソフトウェアの特徴も理解させる。

④ コンピュータの価格と、コンピュータの普及について説明する。
始めは弾道計算などの軍事用目的で開発され、科学技術計算が主体であったが、その後、給与計算などの事務分野に使用されるようになりコンピュータの使用台数が飛躍的に増加した（第２、第３世代）。次にマイクロプロセッサの開発、低価格化によりパソコンが世に出て低価格化の道をたどったので、それまではコンピュータを導入したくとも経済的に導入できなかった企業、個人もコンピュータ（パソコン）を購入できるようになり、コンピュータの使用台数が爆発的に増加した（３．５世代以降）。

⑤ コンピュータの構成について、名称とシステム構成、その特徴を理解させる。

用語

ＥＮＩＡＣ　ＥＤＳＡＣ　ＥＤＶＡＣ　ＵＮＩＶＡＣ Ｉ
真空管　トランジスタ　ＩＣ　ＬＳＩ　ＶＬＳＩ
磁気ドラム　磁気コアメモリ　ブラウン管メモリ　半導体メモリ
固定小数点演算器　浮動小数点演算器　インデックスレジスタ　Ｉ／Ｏプロセッサ
マイクロプログラム　パイプライン処理　キャッシュメモリ　ミニコンピュータ
多重パイプライン　アレイプロセサ　スーパーコンピュータ　マルチプロセッサ
ワークステーション　パーソナルコンピュータ
シンプレックスシステム　デュプレックスシステム　デュアルシステム
マルチプロセッサシステム

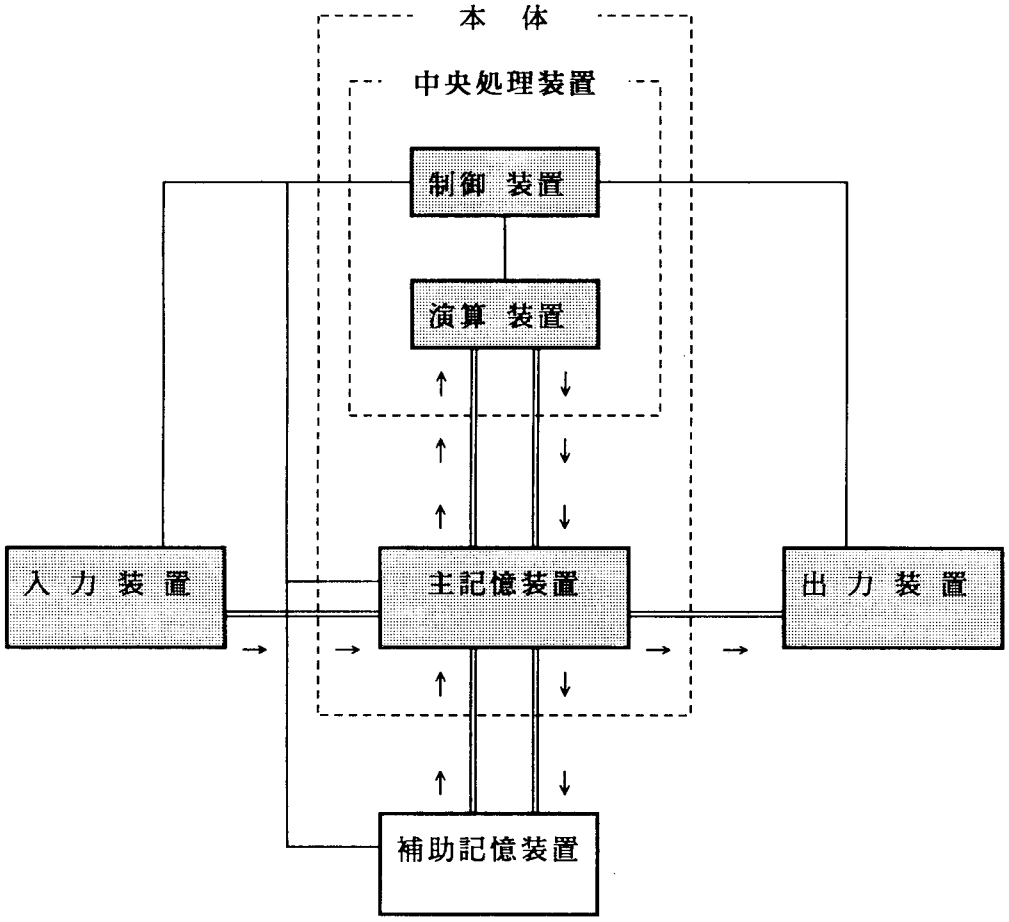
講師ノート

第２種情報処理技術者試験

・コンピュータの発達史に関する問題は頻繁に出題はされるわけではないが、現在のコンピュータがこのようなアーキテクチャになるまでの流れは知っておかなければならない。
・各世代で論理素子、主記憶素子としてどのようなものが使われていたかを理解しておく
・ハードウェア構成は、信頼性の問題と絡んで頻繁に出題されるので、ハードウェアの構成図と説明文から、どのシステムかを解るようにしておく事は必須である。

1．2 基本構造

伝統的なコンピュータは、フォン・ノイマン型と呼ばれ、ストアドプログラム方式（プログラムを主記憶上にストアしておく方式）である。そのノイマン型コンピュータを５つの基本要素に分けると、制御装置、演算装置、記憶装置（主記憶装置＋補助記憶装置）、入力装置、出力装置に分けられ、それぞれの装置の間にデータの移動と制御情報のやりとりがある。



═══　データの流れを示す

――　制御情報の流れを示す

図１．２－１　コンピュータの構成

例）パーソナルコンピュータの機器構成

パーソナルコンピュータを例にそれぞれの機器がどの装置に対応するか説明する。

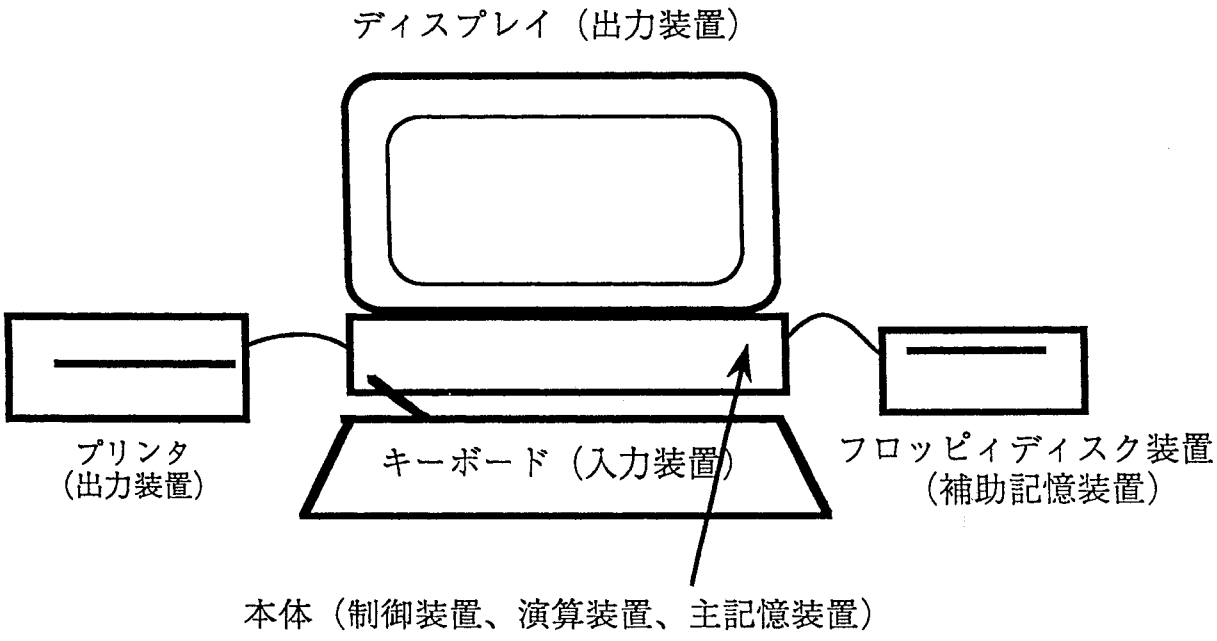


図１．２－２　パーソナルコンピュータの構成

図１．２－１に示されている、制御装置、演算装置、記憶装置、入力装置、出力装置をコンピュータの５大要素という。また、制御装置と演算装置をまとめてＣＰＵ（Central Processing Unit）という。

各装置の機能は以下の通りである。

１．２．１　入力装置（Input Unit）

データを入力する装置で、昔は紙テープ読み取り装置、カード読み取り装置を主に使用していたが、現在では、キーボード、マウスを中心とし、イメージスキャナ、音声入力装置等が入力装置としてあげられる。

１．２．２　出力装置（Output Unit）

データを出力するための装置で、キャラクタディスプレイや、ビットマップディスプレイ、文字を印刷する装置（プリンタ）だけでなく、図形を出力するための装置（Ｘ－Ｙプロッタ）もある。昔は、漢字の出力もできず、速度も遅いシリアルプリンタや、高速ではあるが、印字品質があまり高くないラインプリンタが中心であったが、ワープロやＤＴＰ（デスクトップ・パブリッシング）が発達するにともないプリンタも高品質で、１ページ単位で出力するレーザビームプリンタ（ページプリンタ）が主流になった。

１．２．３　制御装置（Control Unit）

他の各装置を制御する。制御に必要なデータは主記憶装置から読み込み、これを解読し実行する。従って、主記憶装置のどこから読み込むかを示すＰＣ（Program Counter：プログラムカウンタ）、命令レジスタ、アドレスレジスタ、デコーダ（解読器）等からなる。

１．２．４　演算装置（Arithmetic Unit）

四則演算、論理演算（数の大小関係）などを行う。演算装置の機能はＡＬＵ（Arithmetic and Logic Unit）の機能で決まる。演算装置は演算を行うための回路（演算回路）の為のレジスタを複数個持っている。

ＣＰＵの性能を表すための指標として、

ＭＩＰＳ（Mega Instructions Per Second）
１秒間に１００万回（＝2^{20}）命令を実行する。

ＧＩＰＳ（Giga Instructions Per Second）
１秒間に１０億回（＝2^{30}）命令を実行する。

科学技術計算などでは、浮動小数点計算の性能が問題となるため、

ＭＦＬＯＰＳ（Mega FLoating point Operations Per Second）
１秒間に１００万回（＝2^{20}）浮動小数点命令を実行する。

ＧＦＬＯＰＳ（Giga FLoating point Operations Per Second）
１秒間に１０億回（＝2^{30}）浮動小数点命令を実行する。

という値が用いられる。

１．２．５　記憶装置（Storage）

記憶装置には、主記憶装置と補助記憶装置とがある。

主記憶装置は、本体の中にあり制御装置から直接高速に制御されるが、記憶容量としては小さい。以前は磁気コアメモリ等が使用されたが、現在では半導体メモリが主流となっている。アクセス速度を向上するために、キャッシュメモリを使用する場合もある。

補助記憶装置は、主記憶装置の補助となるもので、一般には記憶容量が大きいが、アクセスする速度は遅くなる。補助記憶装置としては、磁気ディスク装置（ハードディスク装置）、フロッピーディスク装置、光磁気ディスク装置、磁気テープ装置、カセットストリーマ装置などがある。

１．２．６　コンピュータで扱う情報の単位

（１）ビット(bit)

コンピュータで扱う情報の最小単位である。オフの状態を０、オンの状態を１とし、０と１だけで表す数が２進数である。この２進数の１桁をビット（bit）と言う。

（２）バイト(byte)

ビットが８個集まったものをバイト（byte）と呼ぶ。
ディスク容量、主記憶容量などを表すための単位に使用される。

１ＫＢ（キロバイト）＝１０２４バイト＝2^{10}バイト
１ＭＢ（メガバイト）＝１０４８５７６バイト＝2^{20}バイト
１ＧＢ（ギガバイト）＝１０７３７４１８２４バイト＝2^{30}バイト

１キロバイトを１０００バイトと表現する事もある。

（３）ワード(word)

バイトまたはビットが複数集まったものをワードと呼ぶ。

ＩＢＭシステム３７０：　１ワード　－　４バイト
インテル８０８６系　：　１ワード　－　２バイト

（４）アドレス(address)

主記憶装置には、バイト、または、ワードごとに固有のアドレスが付けられている。

指導上の留意点

ポイント

① コンピュータを構成する各装置の名称、機能、接続関係を理解する。手持ちのパソコンなどを例として、具体的にどの部分が何という装置かを説明する。以下、抽象的な説明だけでなく、まわりにある具体的な装置を例として説明すると良い。

② データを入力するための入力装置の具体的な機器の概要を理解させる。

③ データを、表示、印刷、保管するための出力装置の具体的な機器の概要を説明する。

④ 制御装置の機能、概要を理解させる。

⑤ 演算装置の概要を説明する。演算速度の性能を示すＭＩＰＳ、ＧＩＰＳ、ＭＦＬＯＰＳ、ＧＦＬＯＰＳなどの用語について説明する。

⑥ 記憶装置の、機能・構成の概要を理解させる。とくに主記憶装置と補助記憶装置との役割分担、具体的な代表的装置を理解させる。

⑦ 制御の流れとデータの流れを混同しないように注意する。

⑧ どのような制御指令に基づき入力装置からからデータが入力されるか、またどのような制御指令に基づき出力装置へデータが出力されるかを理解させる。

⑨ コンピュータで扱う基本的単位（用語）を覚えさせる。

用語

入力装置　出力装置　制御装置　演算装置　記憶装置　ＭＩＰＳ　ＧＩＰＳ
ＭＦＬＯＰＳ　ＧＦＬＯＰＳ　ビット　バイト　ワード　アドレス

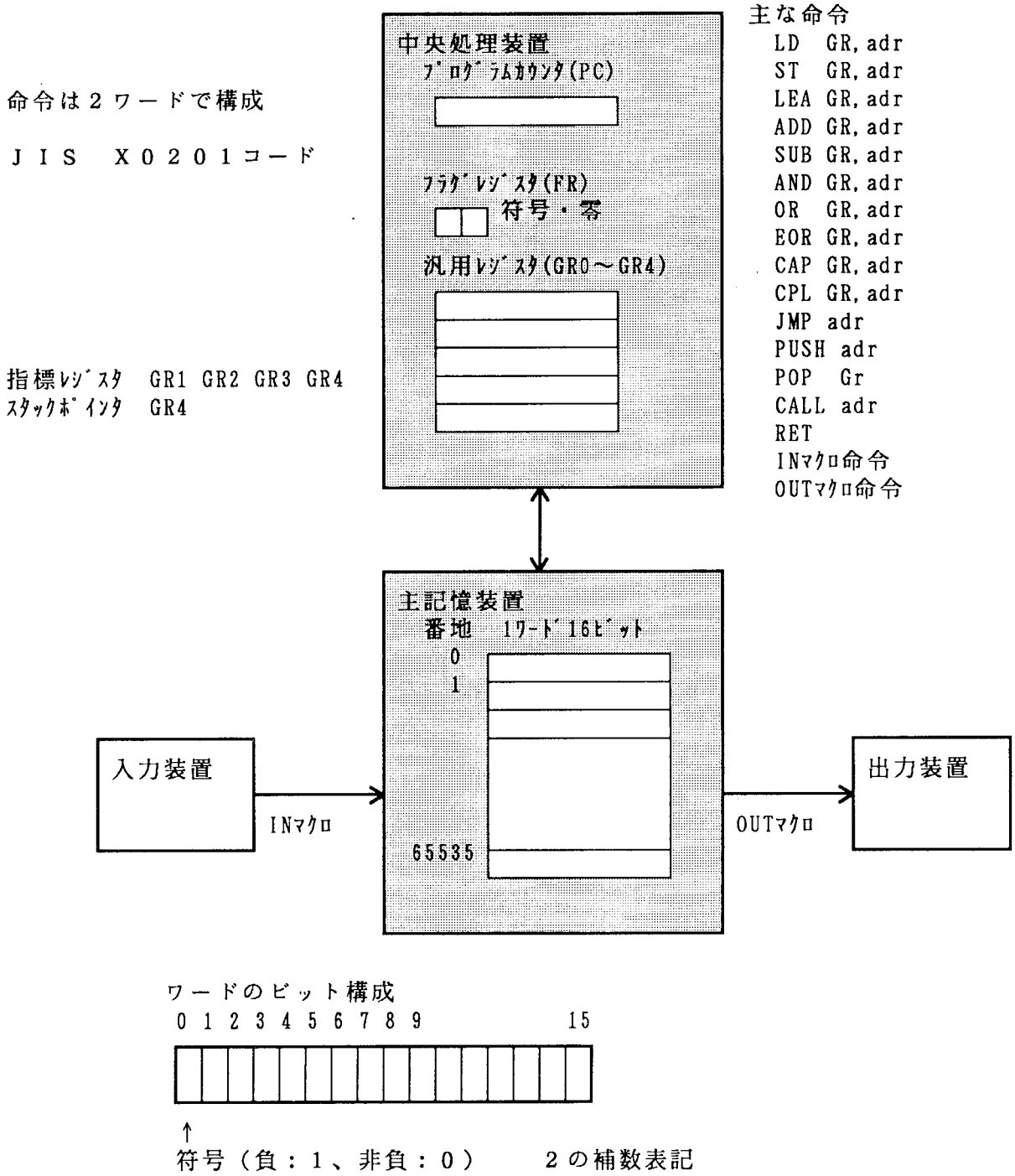
講師ノート

第２種情報処理技術者試験

・計算機構成の問題は頻出はしないが、知っておかなければならない。
・具体的な例として、第２種情報処理技術者試験で出題されるハードウェアとして、「ＣＯＭＥＴ」を例に説明すると良い。次ページにＣＯＭＥＴのハードウェア概要を載せておく。

例）　ＣＯＭＥＴの概要

第２種情報処理技術者試験で出題される「ＣＯＭＥＴ」の仕様を理解しておく。



ワードのビット構成

↑
符号（負：１、非負：０）　　２の補数表記

図１．２－３　ＣＯＭＥＴの構成

1．3　基数

数値の表現の方法には幾通りもあり、基数のとり方により表現が異なる。ここでは、代表的な１０進数、２進数、８進数、１６進数について説明し、基数が示されれば、その基数で表現された値が幾らかを理解できるようにする。
また、基数を指定してやれば、数値の表現をその基数に基づいて表現する能力を養成する。

1．3．1　１０進数（decimal number）

１０進数は、１０を基数とし、０～９までの１０個の数字を用いて全ての数値を表す。２桁以上の１０進数では数字を複数個並べて表現し、各桁には、最下位桁から１（＝10^0）、１０（＝10^1）、１００（＝10^2）・・・の重みを持たせている。例えば、１０進数１２３の各桁の重みは、

$$123 = 1\times10^2+2\times10^1+3\times10^0$$

となる。小数が含まれている場合も同様で、小数点以下の部分は、上位桁から重みが10^{-1}，10^{-2}，となり、１０進数で０．１２３の各桁の重みは

$$0.123 = 1\times10^{-1}+2\times10^{-2}+3\times10^{-3}$$

と表される。

1．3．2　２進数（binary number）

２進数は、２を基数とし、０と１の２個の数字を用いて全ての数値を表す。２桁以上の２進数では数字を複数個並べて表現し、各桁には、最下位桁から１（＝2^0）、２（＝2^1）、４（＝2^2）・・・の重みを持たせている。
一般式で表現すると、２進数$a_n a_{n-1}\cdots a_1 a_0 . a_{-1} a_{-2}\cdots a_{-(m-1)} a_{-m}$は

$$a_n a_{n-1}\cdots a_1 a_0 . a_{-1} a_{-2}\cdots a_{-(m-1)} a_{-m} = a_n\times2^n+a_{n-1}\times2^{n-1}+\cdots+a_1\times2^1+a_0\times2^0+a_{-1}\times2^{-1}+a_{-2}\times2^{-2}+\cdots+a_{-(m-1)}\times2^{-(m-1)}+a_{-m}\times2^{-m}$$

となる。ここで、$0 \leqq a_n, \cdots, a_0, \cdots, a_{-m} < 2$
である。

1．3．3　８進数（octal number）

８進数は、８を基数とし、０～７までの８個の数字を用いて全ての数値を表す。２桁以上の８進数では数字を複数個並べて表現し、各桁には、最下位桁から１（＝8^0）、８（＝8^1）、６４（＝8^2）・・・の重みを持たせている。
一般式で表現すると、８進数$a_n a_{n-1}\cdots a_1 a_0 . a_{-1} a_{-2}\cdots a_{-(m-1)} a_{-m}$は

$$a_n a_{n-1}\cdots a_1 a_0 . a_{-1} a_{-2}\cdots a_{-(m-1)} a_{-m} = a_n\times8^n+a_{n-1}\times8^{n-1}+\cdots+a_1\times8^1+a_0\times8^0+a_{-1}\times8^{-1}+a_{-2}\times8^{-2}+\cdots+a_{-(m-1)}\times8^{-(m-1)}+a_{-m}\times8^{-m}$$

となる。ここで、$0 \leqq a_n, \cdots, a_0, \cdots, a_{-m} < 8$
である。

１．３．４　１６進数（Hexa decimal）

１６進数は、１６を基数とし、０～９までの１０個の数字をと、１０から１５までのかわりにＡ～Ｆの６個の文字を用いて全ての数値を表す。２桁以上の１６進数では数字を複数個並べて表現し、各桁には、最下位桁から１（＝16^0）、１６（＝16^1）、２５６（＝16^2）・・・の重みを持たせている。
一般式で表現すると、１６進数$a_n a_{n-1} \cdots a_1 a_0 . a_{-1} a_{-2} \cdots a_{-(m-1)} a_{-m}$は

$$a_n a_{n-1} \cdots a_1 a_0 . a_{-1} a_{-2} \cdots a_{-(m-1)} a_{-m} =$$
$$a_n \times 16^n + a_{n-1} \times 16^{n-1} + \cdots + a_1 \times 16^1 + a_0 \times 16^0 + a_{-1} \times 16^{-1}$$
$$+ a_{-2} \times 16^{-2} + \cdots + a_{-(m-1)} \times 16^{-(m-1)} + a_{-m} \times 16^{-m}$$

となる。ここで、$0 \leqq a_n, \cdots, a_0, \cdots, a_{-m} < 16$
である。

１．３．５　基数同士の対応関係

１０進数、２進数、８進数、１６進数の関係をまとめると、以下のようになる。

１０進数	２進数	８進数	１６進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

表１．３－１　１０進数、２進数、８進数、１６進数の対応関係

１．３．６　基数の変換

（１）２進数→１０進数、１０進数→２進数への変換

（ａ）２進数→１０進数

２進数の各桁ごとに数値と重みを掛けて得られる各数値を加算し、結果を得る。

例　２進数１０１．０１１を１０進数に変換する

$$\begin{array}{llllllll} 2^{2} & 2^{1} & 2^{0} & & 2^{-1} & 2^{-2} & 2^{-3} & \leftarrow \text{各桁の重み} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & \\ (1 & 0 & 1 & . & 0 & 1 & 1)_{2} & \end{array}$$

$$\begin{aligned} &= 1\times2^{2}+0\times2^{1}+1\times2^{0}+0\times2^{-1}+1\times2^{-2}+1\times2^{-3} \\ &= 1\times4+0\times2+1\times1+0\times1/2+1\times1/4+1\times1/8 \\ &= 4+0+1+0+1/4+1/8 \\ &= 4+1+0.25+0.125 \\ &= 5.375 \end{aligned}$$

（ｂ）１０進数→２進数

①整数部

1)１０進数の整数部を２進数の基数である２で割り、商と余り（０または１）を求める。

2)上で得られた商をさらに２で割り、商と余りを求める。これを商が０になるまで繰り返す。

3)計算が終了したら、余りを得た順と逆順序に上位から並べる。

4)これが求める２進数である。

例　１０進数１９を２進数にする。

	（商）		余り		
2)	19				１９を２で割る
2)	9	…….	1	↑	商９余り１，さらに９を２で割る
2)	4	…….	1	↑	商４余り１，さらに４を２で割る
2)	2	…….	0	↑	商２余り０，さらに２を２で割る
2)	1	…….	0	↑	商１余り０，さらに１を２で割る
	0		1	↑	商０余り１．

求める２進数は、１００１１となる。

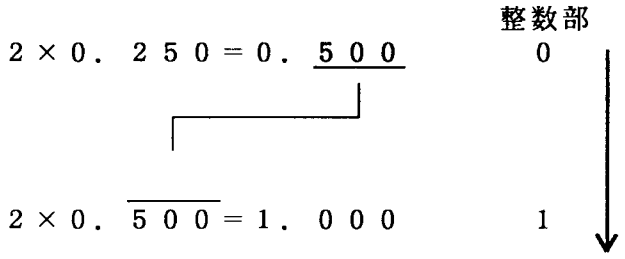
②小数部

1)１０進数の小数部に２進数の基数である２を掛けて積を求める。この積の整数部（０または１）が２進数の小数部の１桁になる。

2)1)で得られた積の小数部のみを新しい１０進数として積の小数部が０になるまで小数部分を２倍する。

3)計算が終了したら整数部１、０を得た順に上位から並べると求める小数部の２進数表現が得られる。

例　１０進数０．２５０を２進数にする。

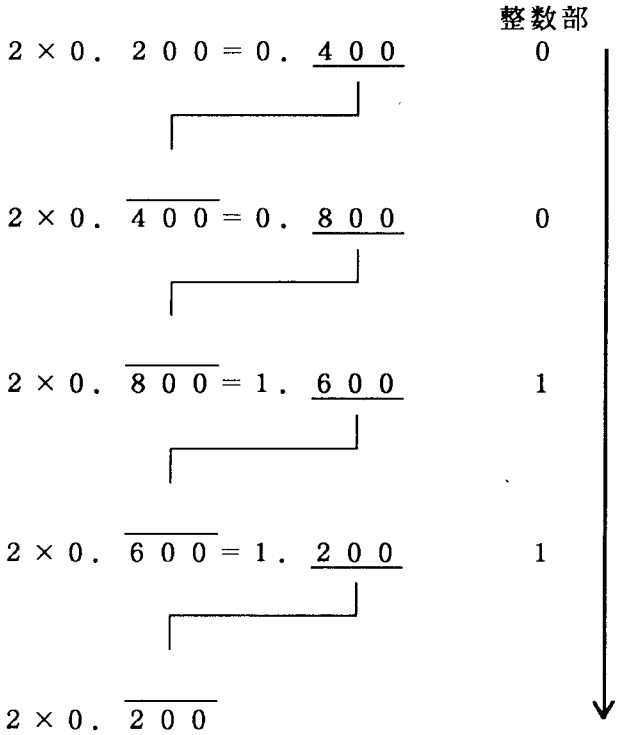


得られた整数部を上位より並べると　０１となる。従って

（０．２５０）$_{10}$　＝　（０．０１）$_{2}$

となる。

例　１０進数０．２を２進数にする（循環小数になる例）



ここで、小数部が０．２となり、循環小数となる事がわかる。

そこで、ここまでで得られた整数部を上位より並べると

００１１

となる。従って

（０．２００）$_{10}$　＝　（０．０００１１）$_{2}$

となる。

（２）８進数→１０進数、１０進数→８進数

（ａ）８進数→１０進数

８進数の各桁ごとに数値と重みを掛けて得られる各数値を加算し、結果を得る。

例　８進数２７５．１５を１０進数に変換する

$8^2\ 8^1\ 8^0\quad 8^{-1}\ 8^{-2}$　　← 各桁の重み
↓　↓　↓　　↓　↓
（２　７　５　．　１　５）$_8$

$= 2\times8^2+7\times8^1+5\times8^0+1\times8^{-1}+5\times8^{-2}$
$= 2\times64+7\times8+5\times1+1\times1/8+5\times1/64$
$= 128+56+5+0.125+0.078125$
$= 189.20312$

（ｂ）１０進数→８進数

①整数部

1)１０進数の整数部を８進数の基数である８で割り、商と余りを求める。

2)上で得られた商をさらに８で割り、商と余りを求める。これを商が０になるまで繰り返す。

3)計算が終了したら、余りを得た順と逆順序に上位から並べる。

4)これが求める８進数である。

例　１０進数１４７を８進数にする。

	（商）	余り		
8）	１４７			１４７を８で割る
8）	１８	……. 3	↑	商１８余り３，さらに１８を８で割る
8）	２	……. 2	↑	商２余り２，さらに２を８で割る
	０	2	↑	商０余り２．

求める８進数は、２２３となる。

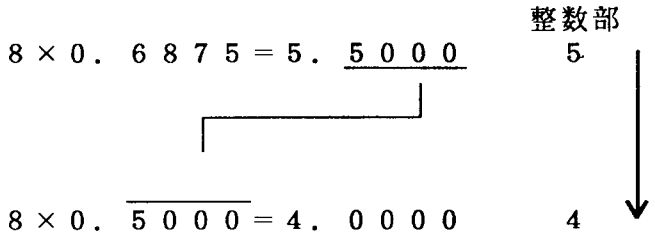
②小数部

1)１０進数の小数部を８進数の基数である８を掛けて積を求める。この積の整数部が８進数の小数部の１桁になる。

2)1)で得られた小数部のみを新しい１０進数として積の小数部が０になるまで小数部分を８倍する。

3)計算が終了したら整数部を得た順に上位から並べると求める小数部の８進数表現が得られる。

例　１０進数０．５６２５を８進数にする。



得られた整数部を上位より並べると　５４となる。従って

（０．６８７５）$_{10}$　＝　（０．５４）$_{8}$

となる。

（３）１０進数→１６進数、１６進数→１０進数

（ａ）１６進数→１０進数

１６進数の各桁ごとに数値と重みを掛けて得られる各数値を加算し、結果を得る。

例　１６進数Ｄ３Ａ．Ｃ５を１０進数に変換する

16^{2}　16^{1}　16^{0}　16^{-1}　16^{-2}　← 各桁の重み

（　Ｄ　３　Ａ　．　Ｃ　５）$_{8}$

＝１３×16^{2}＋３×16^{1}＋１０×16^{0}＋１２×16^{-1}＋５×16^{-2}

＝１３×２５６＋３×１６＋１０×１＋１２×１／１６＋５×１／２５６

＝３３２８　＋　４８　＋　１０　＋０．７５　＋０．０１９５３１２

＝　３３８６．７６９５３１

（ｂ）１０進数→１６進数

①整数部

1)１０進数の整数部を１６進数の基数である１６で割り、商と余りを求める。

2)上で得られた商をさらに１６で割り、商と余りを求める。これを商が０になるまで繰り返す。

3)計算が終了したら、余りを得た順と逆順序に上位から並べる。

4)これが求める１６進数である。

例　１０進数３５１を１６進数にする。

	（商）	余り	
１６）３５１			３５１を１６で割る
１６）２１	･･･	１５ ↑	商２１余り１５，さらに２１を１６で割る
１６）１	･･･	５ ↑	商１余り５，さらに１を１６で割る
０		１ ↑	商０余り１．

求める１６進数は、１５Ｆとなる。

②小数部

1)１０進数の小数部を１６進数の基数である１６倍を掛けて積を求める。この積の整数部が１６進数の小数部の１桁になる。

2)1)で得られた小数部のみを新しい１０進数として積の小数部が０になるまで小数部分を１６倍する。

3)計算が終了したら整数部を得た順に上位から並べると求める小数部の１６進数表現が得られる。

例　１０進数０．７８９０６２５を１６進数にする。

整数部

１６×０．７８９０６２５＝１２．６２５０　　１２　（Ｃ）$_{16}$

↓

１６×０．６２５０００００＝１０．００００　　１０　（Ａ）$_{16}$

得られた整数部を上位より並べると　ＣＡとなる。従って

（０．７８９０６２５）$_{10}$　＝　（０．ＣＡ）$_{16}$

となる。

（４）２進数→８進数、　２進数→１６進数

（ａ）２進数→８進数への変換

２進数から８進数へ変換する場合は、小数点の位置を基準にして上位の桁、下位の桁をそれぞれ３桁づつ区切り、その２進数３桁の数字を８進数字（０～７）に置き換える。なお、２進数の桁数が３桁にならない場合には上位の桁には前に０を、下位桁には後ろに０を付け加えて置き換える。

例）２進数１１０１１０１．１１０１１を８進数に変換する。

００１	１０１	１０１	．	１１０	１１０
1	5	5		6	6

答　（１５５．６６）$_{8}$

（ｂ）２進数→１６進数への変換

２進数から１６進数へ変換する場合も同様に、小数点の位置を基準にして上位の桁、下位の桁をそれぞれ４桁づつ区切り、その４桁の数字を１６進数字（０～９，Ａ～Ｆ）に置き換える。なお、２進数の桁数が４桁にならない場合には上位桁には前に０を、下位桁には後ろに０を付け加えて置き換える。

例）２進数１０１１１１０１１０１．１１０１１を１６進数に変換する。

０１０１	１１１０	１１０１	．	１１０１	１０００
5	E	D		D	8

答（５ＥＤ．Ｄ８）$_{16}$

（５）８進数→２進数、１６進数→２進数

（ａ）８進数→２進数への変換

８進数から２進数へ変換する場合は、８進数の各桁を３桁の２進数で置き換えれば良い。

例）８進数７６３．２５を２進数に変換する。

7	6	3	.	2	5
111	110	011		010	101

答　（１１１１１００１１．０１０１０１）$_2$

（ｂ）　１６進数→２進数への変換

１６進数から２進数へ変換する場合は、１６進数の各桁を４桁の２進数で置き換えれば良い。

例）１６進数１ＦＣ．１Ｂを２進数に変換する。

1	F	C	.	1	B
0001	1111	1100		0001	1101

答（１１１１１１１００．０００１１１０１）$_2$

（６）８進数→１６進数、１６進数→８進数

２進数に一度変換してから行う。即ち、８進数から１６進数の変換は、８進数を３桁の２進数の列に変換し、それを４桁の２進数の集まりに再構成し、それを１６進数に直す。

他方、１６進数から８進数の変換は、１６進数を４桁の２進数の列に変換し、それを３桁の２進数の集まりに再構成し、それを８進数に直す。

例）８進数７６３．５２を１６進数に変換する。

7	6	3		5	2
111	111	001	1 .	101	0100
1	F	3		A	8

答　（１Ｆ３．Ａ８）$_{16}$

１．３．７　各種基数での演算

（１）加算

Ｎ進数では、最大の数が（Ｎ－１）で、各桁の和がＮ以上になったら１つ上位の桁に桁あがりを加え、元の桁には（和－Ｎ）を残す。

（ａ）２進数の加算

（１０１０１１）$_2$＋（１１１０１０）$_2$　を求める。

```
      1  0  1  0  1  1
+)    1△1△1  0△1  0          △桁上げを示す
  1  1  0  0  1  0  1
```

（ｂ）８進数の加算

（７７６）$_8$＋（５０３）$_8$　を求める。

```
      7  7  6
+)    5△0△3          △桁上げを示す
  1  5  0  1          桁上げ
                        ↓
   ┌──────────→ (6 + 3 + 0) − 8 = 1
   │  ┌───────→ (7 + 0 + 1) − 8 = 0
   │  │  ┌────→ (7 + 5 + 1) − 8 = 5
```

（ｃ）１６進数の加算

（Ｅ０Ｃ）$_{16}$＋（ＡＢ７）$_{16}$　を求める。

```
      E  0  C
+)    A  B△7          △桁上げを示す
  1  8  C  3                  桁上げ
                               ↓
   ┌──────────→ (12 +  7 +  0) − 16 = 3
   │
   │
   └──────────→ (14 + 10 +  0) − 16 = 8
```

（２）減算

加算と同様に基数の違いを考慮し、１０進数の場合と同じようにして演算を行う。ある桁の減算で２つの数の演算結果が０よりも小さいときは１桁上の位から１を借りてきて演算を行う。

（ａ）２進数の減算

（１０１０１１１）$_2$－（１１１０１０）$_2$ を求める。

```
    1  0  1  0  1  1  1
-)     1△1△1  0  1  0
    ---------------------
       1  1  1  0  1
```

△借りが生じた事を示す

（ｂ）８進数の減算

（７３４）$_8$－（５３６）$_8$ を求める。

```
    7  3  4
-)  5△3△6
    ---------
    1  7  6
```

△借りが生じた事を示す

（ｃ）１６進数の減算

（Ｅ０Ｃ）$_{16}$－（ＡＢＦ）$_{16}$ を求める。

```
    E  0  C
-)  A△B△F
    ---------
    3  4  D
```

△借りが生じた事を示す

（３）乗算

２進数に変換してから演算を行う。

２進数の乗算

（１０１１）$_2$×（１１０）$_2$を求める

```
          1  0  1  1
    ×)          1  1  0
    ---------------------
          0  0  0  0
       1  0  1  1
    1  0  1  1
    ---------------------
 1  0  0  0  0  1
```

（４）除算

２進数に変換してから除算を行う。

２進数の除算

（１０１１１０）$_2$÷（１１０）$_2$　を行う

```
          1  1  1        ・・・・　商
    ---------------
1 1 0) 1 0 1 1 1 0
         1 1 0
       -------
         1 0 1 1
           1 1 0
       -------
           1 0 1 0
             1 1 0
       -------
             1 0 0        ・・・・　余り
```

指導上の留意点

ポイント

① １０進数、２進数、８進数、１６進数のそれぞれによる数値の表現を理解させる。特に１６進数は、Ａ～Ｆという文字を１０進数で１０～１５までの数字と見なすという、新しい考え方を学ぶので、学生はなかなか慣れないかも知れないが、
Ａ→１０、Ｂ→１１、Ｃ→１２、Ｄ→１３、Ｅ→１４、Ｆ→１５
だということを繰り返し説明して理解させる事が重要である。

② 各基数同士の変換の考え方、方法を理解させ、実際に変換を行わせる。小数点以下をもつ数の変換も行わせ、１０進数では割り切れる数が２進数では循環小数となり、割り切れない数もある事を理解させる。実際に例題を解かせる事により理解を深める。

③ 各基数での演算（加減乗除）の考え方を理解させる。乗算、除算では、１０進数、２進数以外は、２進数に変換してから演算を行わせる。
加算の桁上がり、減算の借りに注意する。１６進数の加算、減算はなかなか慣れないので、充分に練習しておく必要がある。

用語

基数　１０進数　２進数　８進数　１６進数

講師ノート

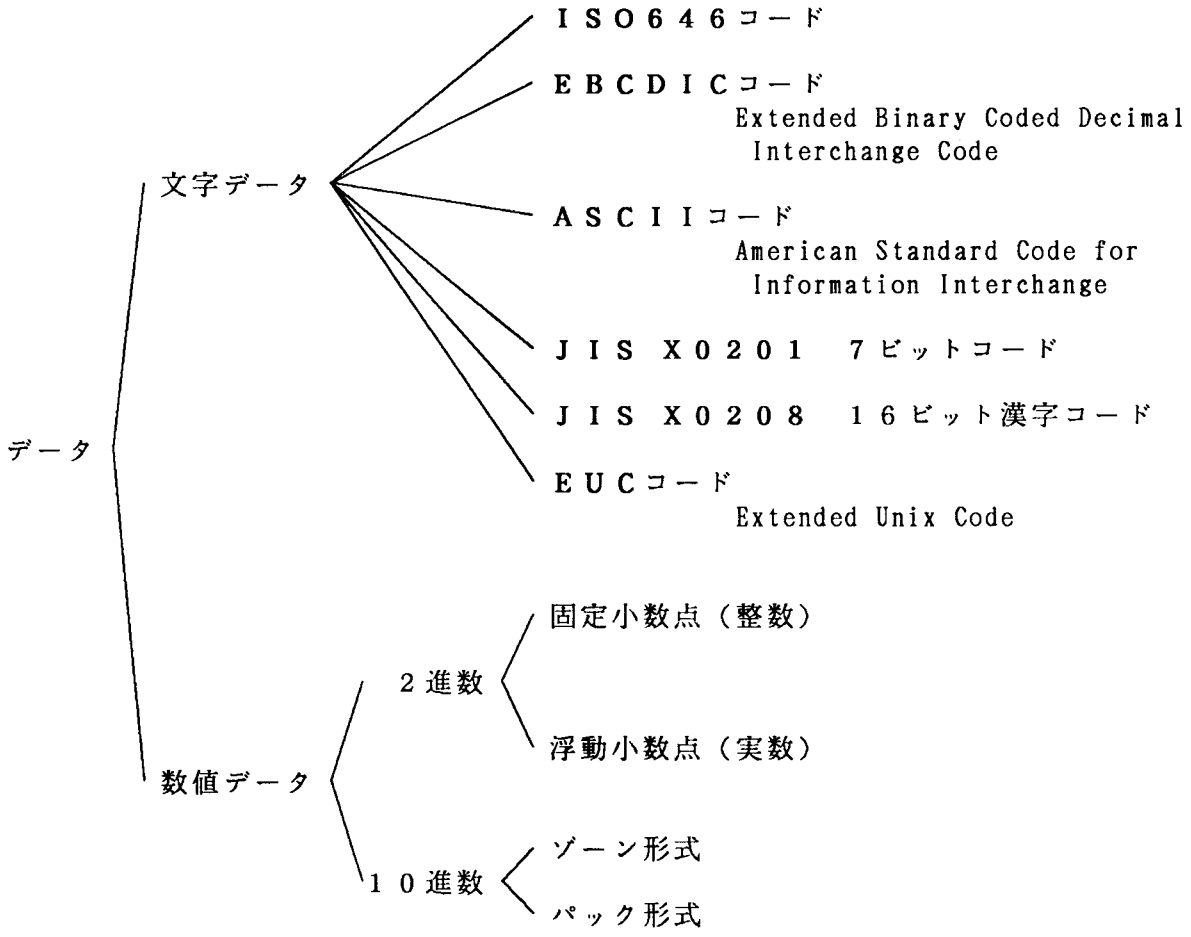
第２種情報処理技術者試験

・数値の表現に関する問題は、しばしば出題されるので、充分に理解しておく事が重要である。

・特に１６進数の演算は、様々なところで利用するので、充分に練習問題をこなしておく事が重要である。

１．４　コード体系

コンピュータで使用されるデータを分類すると、次のようになる

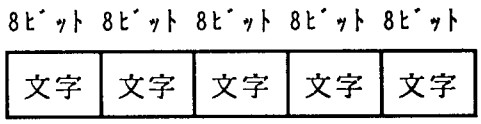


文字データについては１．４節で、数値データについては１．５節で学習する。

１．４．１　文字のデータ表現

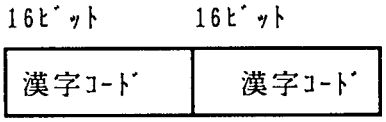
（１）　英字（アルファベット、特殊文字）、数字、カナ文字

通常の英字、数字、カナ文字、特殊記号などキーボードのキーに割り振られている文字は、８ビットで表現されている。



（２）　漢字

漢字は、種類が多いため８ビットでは表現できず１６ビットで１文字を表現する。



（３）　１バイトで表せない文字

漢字以外にも、ハングル、ヘブライ語など１バイトでは表せない文字は、世界にはたくさんある（２５６種類以上の文字があれば１バイトでは表せない）。これを処理するために１文字を１６ビットで表現し、各国の文字が扱えるようになりつつある。これを支援するオペレーティング・システムも開発されはじめている。（例えばマッキントッシュ　Ｓｙｓｔｅｍ７）

１．４．２　ＥＢＣＤＩＣコード（Extended Binary Coded Decimal Interchange Code）

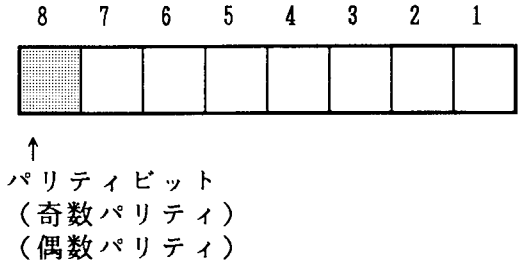
ＩＢＭが開発した、８ビットで１文字を表すコードである。
汎用機の世界では標準的なコード体系である。

行＼列	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DEL	DS		SP	&	＿			ソ			{	}	＼	0
1	SOH	DC1	SOS		。	ェ	／		ア	タ			A	J		1
2	STX	DC2	FS	SYN	「	ォ			イ	チ	へ		B	K	S	2
3	ETX	TM			」	ャ			ウ	ツ	ホ		C	L	T	3
4	PF	RES	BYP	PN	、	ュ			エ	テ	マ		D	M	U	4
5	HT	NL	LF	RS	・	ョ			オ	ト	ミ		E	N	V	5
6	LC	BS	ETB	UC	ヲ	ッ			カ	ナ	ム		F	O	W	6
7	DEL	IL	ESC	EOT	ァ				キ	ニ	メ		G	P	X	7
8	GE	CAN			ィ	―			ク	ヌ	モ		H	Q	Y	8
9	RLF	EM			ゥ				ケ	ネ	ヤ		I	R	Z	9
A	SMM	CC	SM		¢	!			コ	ノ	ユ	レ				
B	VT	CU1	CU1	CU3	.	¥		#				ロ				
C	FF	IFS		DC4	<	*	%	@	サ		ヨ	ワ				
D	CR	IGS	EMQ	NAK	(	)	−	|	シ	ハ	ラ	ン				
E	SO	IRS	ACK		+	:	>	=	ス	ヒ	リ	゛				
F	SI	IUS	BEL	SUB	|	¬	?	”	セ	フ	ル	°				EO

表１．４－１　ＥＢＣＤＩＣコード

１．４．３　ＡＳＣＩＩコード（American Standard Code for Information Interchange）

ＡＳＡ（アメリカ規格協会）よりＩＳＯ（国際標準化機構）に提出され、アメリカで最初に定められた標準コードである。ＩＳＯ６４６コード（７ビットコード）にパリティビットを付加したコードである。ＪＩＳコードの母体となっている。



列／行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DE		0	@	P		p								
1	SH	D1	!	1	A	Q	a	q								
2	SX	D2	"	2	B	R	b	r								
3	EX	D3	#	3	C	S	c	s								
4	ET	D4	$	4	D	T	d	t								
5	EQ	NK	%	5	E	U	e	u								
6	AK	SN	&	6	F	V	f	v								
7	BL	EB	'	7	G	W	g	w								
8	BS	CN	(	8	H	X	h	x								
9	HT	EM	)	9	I	Y	i	y								
A	LF	SB	*	:	J	Z	j	z								
B	HM	EC	+	;	K	[	k	{								
C	CL	→	,	<	L	＼	l	|								
D	CR	←	-	=	M	]	m	}								
E	SO	↑	.	>	N	^	n	~								
F	SI	↓	/	?	O	_	o									

表１．４－２　ＡＳＣＩＩコード

１．４．４　ＪＩＳ　Ｘ０２０１　コード

７ビット、８ビット、の２種類があり、ＩＳＯの標準規格（ＩＳＯ　６４６）コードにカナコードを付け加えたもの。わが国の標準コードとなっている。

列 / 行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DEL	SP	0	@	P	、	p				ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
5	END	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	、	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	FF	FS	'	<	L	¥	l	|			ャ	シ	フ	ワ		
D	CR	GS	-	=	M	]	m	}			ュ	ス	ヘ	ン		
E	SO	RS	.	>	N	^	n	-			ョ	セ	ホ	゛		
F	SI	US	/	?	O	_	o	DEL			ッ	ソ	マ	゜		

表１．４－３　ＪＩＳ　Ｘ０２０１コード

１．４．５　漢字コード（ＪＩＳ　Ｘ０２０８）

日本語の処理で基本となるコード体系で、６３５３種（第１水準２９６５文字、第２水準３３８８文字）の漢字を始め、英数字、仮名、記号、罫線素片など合計６８７７種の文字からなる２バイトコードである。

ＩＳＯ２０２２に従った複数バイトＧ０集合の１つとして位置づけられ、ＪＩＳＸ２０１が規定する英数字・カタカナ用の１バイトコードと併用できる。

ここで、Ｇ０集合とは、７単位符号（ＪＩＳ　Ｘ０２０１）を拡張するときの文字集合で、９４個の図形文字の集合と、複数バイト集合を含む。

漢字コードの割り付けを、シフトＪＩＳ体系で示す。

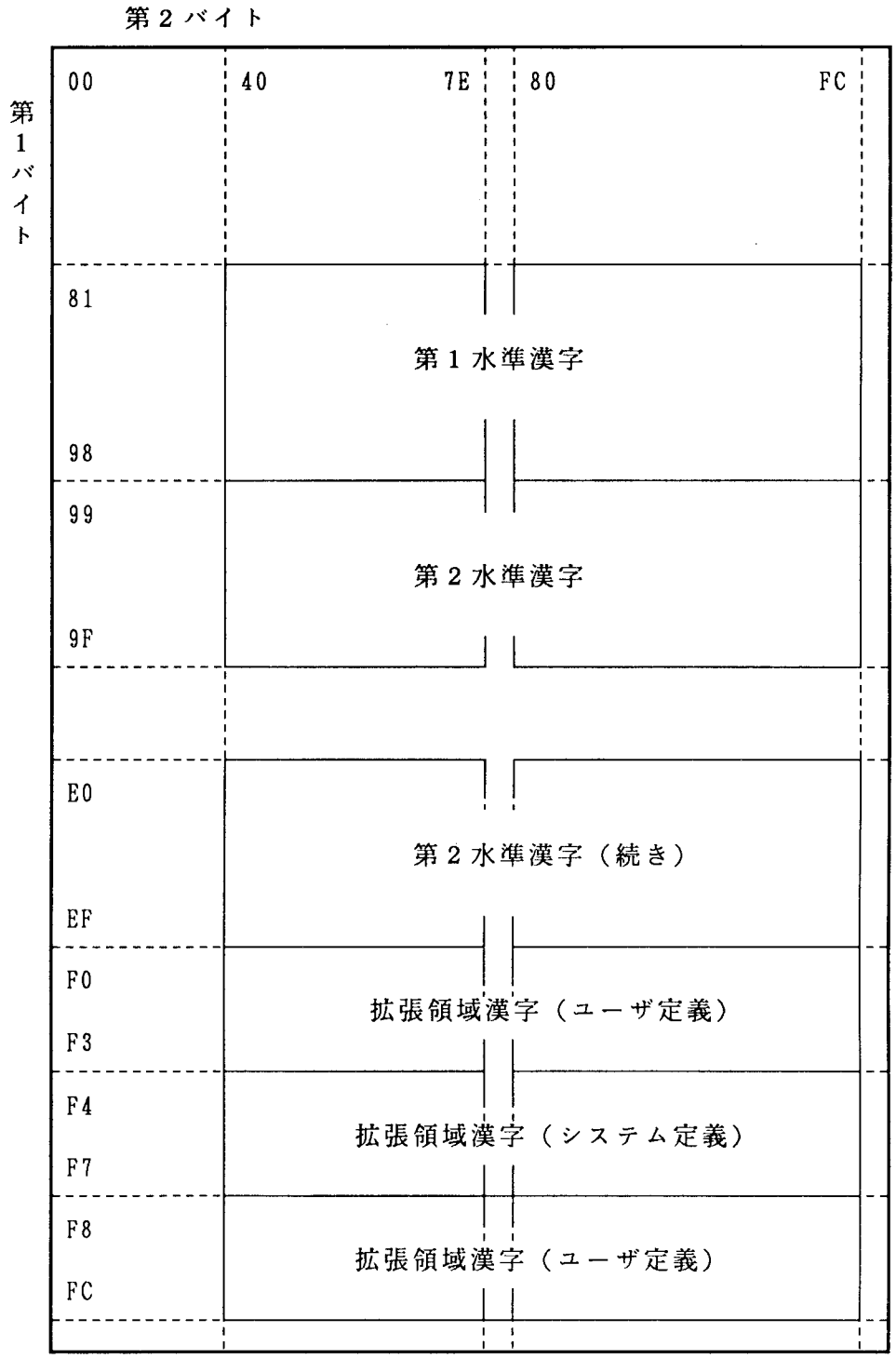


表１．４－４　シフトＪＩＳ体系

１．４．６　ＥＵＣコード（Extended Unix Code）

ＵＮＩＸで使用されているコードである。
ＥＵＣは必ずＡＳＣＩＩ文字集合が割り当てられる１つの基本コード・セット（コード・セット０）と、ユーザが選択する文字集合に割り当てる事ができる３つの補助コード・セット（コード・セット１～３）で構成される。
コード・セットはＥＵＣ表現の最上位ビット（ＭＳＢ）の値とシングルシフト文字（ＳＳ２，ＳＳ３）で４つのコード・セットが表現される。

１バイトの基本コード・セット（コード・セット０）を表すには、ＭＳＢを０にする。
コード・セット１を表すには各バイトのＭＳＢを１にする。
コード・セット２、コード・セット３は、各バイトのＭＳＢを１にするほかにＳＳ２（１６進数８Ｅ）、ＳＳ３（１６進数８Ｆ）を付加する事により区別する。

日本語処理に必要な半角カナは、コード・セット１の１バイト文字、漢字は２バイト文字に割り当てて使用している。

		ＥＵＣによる表現（ビット形式）	
1	コード・セット０	0xxxxxxx	１バイト文字
2	コード・セット１	1xxxxxxx 1xxxxxxx　1xxxxxxx ・ ・	１バイト文字 ２バイト文字 ３バイト文字
3	コード・セット２ SS2：ｼﾝｸﾞﾙｼﾌﾄ2 （８Ｅ）	SS2　1xxxxxxx SS2　1xxxxxxx　1xxxxxxx ・ ・	１バイト文字 ２バイト文字 ３バイト文字
4	コード・セット３ SS3：ｼﾝｸﾞﾙｼﾌﾄ3 （８Ｆ）	SS3　1xxxxxxx SS3　1xxxxxxx　1xxxxxxx ・ ・	１バイト文字 ２バイト文字 ３バイト文字

表１．４－５　ＥＵＣコード体系

指導上の留意点

ポイント

① コンピュータは、文字を図形（パターン）として記憶しておくのではなく、コード化して記憶している事を説明する。何故そのようにしているかについて（コード化の利点について）説明する。

② 各コード体系の特徴、使用されているマシン・分野による違いを理解させる。あるコードが別のコード体系ではどのように表現されるか（コード変換の問題）を理解させる。例えば、ＩＢＭマシンで作成したデータをＵＮＩＸマシンで処理しようとした場合に、何をしなければならないかを具体的に説明する。コード変換を行う場合、行わない場合（文字化けする）と、実際のマシンを使用して説明できると良い。

③ ２バイトコード（１６ビットで１文字を表す）を支援するオペーレティングシステムが出てきた事を説明し、その利害（ハングルやヘブライ語が扱える、反面、従来８ビットで表現できたものはどう扱うか）を説明する。

用語

ＥＢＣＤＩＣコード　ＡＳＣＩＩコード　ＩＳＯ６４６コード
ＪＩＳ Ｘ０２０１コード　ＪＩＳ Ｘ０２０８コード　ＥＵＣコード

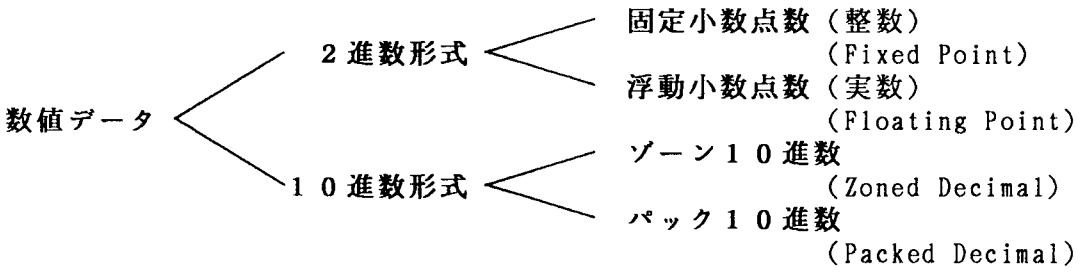
講師ノート

第２種情報処理技術者試験

・コード体系の問題が単独で出題される事はないが、是非知っておかねばならない事項である。

１．５　数値の表現

コンピュータの内部で利用される数値データの表現形式は、次の４通りがある。



１．５．１　固定小数点

通常コンピュータ上で数値を表すときは、表示された数値の右端に小数点があるものとして考える。これを固定小数点と呼ぶ。

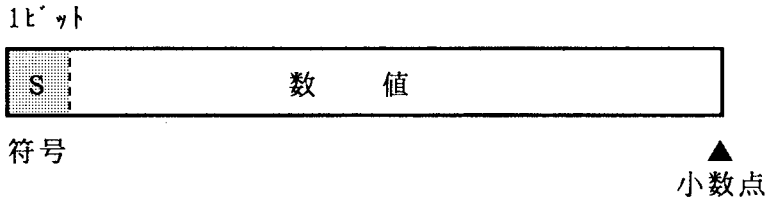
固定小数点形式で、負の数を表す場合には次の３通りの方法がある。

- **符号ビットと数値の絶対値を用いる方法**
- **１の補数を用いる方法**
- **２の補数を用いる方法**

（１）符号ビットと数値の絶対値を用いる方法

２進数の先頭の１ビットを符号ビットとし、符号ビットが０ならば正の数、１ならば負の数を表す事とし、残りのビットで数値の絶対値を表す。

浮動小数点の仮数部などの表現に、この方法が使われている。



例）１０進数の１４と－１４を２進数８ビットで表現してみる。

（１４）$_{10}$　　００００１１１０
↑
先頭符号ビット（正）

（－１４）$_{10}$　　１０００１１１０
↑
先頭符号ビット（負）

（２）補数（１の補数、２の補数）を用いる方法

補数は、２進数の減算を行う場合や、浮動小数点数の指数部の負数を表現する場合に用いられる。
２進数には、２の補数と１の補数の２種類がある。いま、桁数をｎとしたとき、

Ｎの２の補数は、$2^n - N$を２進数で表示したもの、
Ｎの１の補数は、$2^n - 1 - N$を２進数で表示したものと定義する。

例）桁数８では、１２＝（００００１１００）$_2$の１の補数は
$2^n - 1 - N = 2^8 - 1 - 12 = (243)_{10}$
$= (11110011)_2$

２の補数は、
$2^n - N = 2^8 - 12 = (244)_{10}$
$= (11110100)_2$
となる。

1) １の補数の求め方

①　任意の数Ｎを２進数表示で記述する。
②　桁数が足りない場合は、上位桁に０を追記する。
③　２進数の各桁の０と１とを書き換える。

以上で得られた２進数が１の補数である。

例）（１５）$_{10}$の１の補数を８桁で求める。

①　（１５）$_{10}$を２進数表示で表すと　　１１１１
②　桁数８で表すと　　００００１１１１
　　↓↓↓↓↓↓↓↓
③　０と１を書き換えると　　１１１１０００１

以上の結果求める値は（１１１１０００１）$_2$となる

2) ２の補数の求め方

①から③までは１の補数の求め方と同じである。

④　③で得られた結果に１を加える。

以上で得られた２進数が２の補数である。

例）（１９）$_{10}$の２の補数を８桁で求める。

①　（１９）$_{10}$を２進数表示で表すと　　１００１１
②　桁数８で表すと　　０００１００１１
　　↓↓↓↓↓↓↓↓
③　０と１を書き換えると　　１１１０１１００

④　１を加えると　　１１１０１１０１

以上の結果求める値は（１１１０１１０１）$_2$となる。

4桁で、1の補数と2の補数を表すと表1．5－1のようになる。ここで注意すべき事は、1の補数では0の表現が（0000）$_2$と（1111）$_2$の2種類存在する事と、－8を4桁では表現できない事など2の補数表示に比べて不都合が多くなるので現在では負数の表現には2の補数を用いている。

表1．5－1　1および2の補数

10進数	1の補数	2の補数	10進数	1の補数	2の補数
0	0000	0000	－0	1111	0000
1	0001	0001	－1	1110	1111
2	0010	0010	－2	1101	1110
3	0011	0011	－3	1100	1101
4	0100	0100	－4	1011	1100
5	0101	0101	－5	1010	1011
6	0110	0110	－6	1001	1010
7	0111	0111	－7	1000	1001
8			－8		1000

（3）補数を用いた減算

減算を、補数を用いて加算で計算する事を考える。いま、減算（25）$_{10}$－（17）$_{10}$を8桁で補数を用いて計算する。即ち、（25）$_{10}$－（17）$_{10}$＝（25）$_{10}$＋（－17）$_{10}$の計算を行う。

25を8桁の2進数で表す	00011001
17を8桁の2進数で表す	00010001
17の1の補数	11101110
17の2の補数	11101111

①　1の補数を用いて計算する

```
      00011001
+)    11101110
  1← 00000111
  桁あふれ
```

8桁で考えた場合、上位への桁あふれは無視するが、1の補数の計算では最後に足して計算する。従って、

00000111＋1＝（1000）$_2$＝（8）$_{10}$

となる。

② ２の補数を用いて計算する

```
      0 0 0 1 1 0 0 1
+)    1 1 1 0 1 1 1 1
  1← 0 0 0 0 1 0 0 0
  桁あふれ
```

２の補数の計算では、上位への桁あふれは無視する。従って、

$(0\,0\,0\,0\,1\,0\,0\,0)_2 = (8)_{10}$

となる。

② 計算結果が負となる例

減算 $(17)_{10} - (25)_{10}$ を８桁で２の補数を用いて計算する。即ち、$(17)_{10} - (25)_{10} = (17)_{10} + (-25)_{10}$ の計算を行う。

```
      0 0 0 1 0 0 0 1      １７
+)    1 1 1 0 0 1 1 1      ２５の２の補数
      1 1 1 1 1 0 0 0      最上位が１のため負数を表す
-                   1      １を減じる
      1 1 1 1 0 1 1 1      ０、１を書き換えて元の数を求める
      ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
      0 0 0 0 1 0 0 0
```

$(1\,0\,0\,0)_2 = (8)_{10}$ 従って、$(1\,1\,1\,1\,0\,1\,1\,1)_2 = (-8)_{10}$

となる。

（４）数値の表す範囲

正負の符号を考えないｎ桁の２進数で表現できる数値Ｘは

$0 \leqq X \leqq 2^n - 1$ の範囲で表現され、

最小値＝０、最大値＝$2^n - 1$、表現できる数の種類は2^nとなる。

負の数も考えた、ｎ桁の２進数で表現できる数値Ｙは、

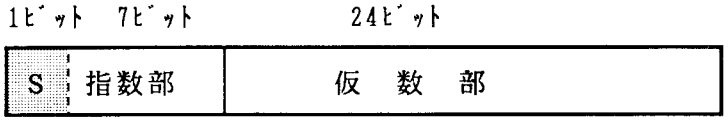
$-2^{n-1} \leqq Y \leqq 2^{n-1} - 1$ の範囲で表現され、

最小値＝-2^{n-1}、最大値＝$2^{n-1} - 1$、表現できる数の種類は2^nとなる。

１．５．２　浮動小数点

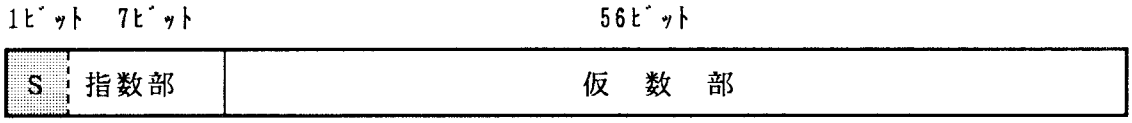
浮動小数点数　$N = M \times B^{E}$　とおくと、コンピュータ内では、１語長３２ビットの単精度の場合では、（１）のように先頭１ビットが仮数部の符号（正または零：０、負：１）、指数部は７ビット、仮数部に２４ビットが割り当てられる。また、６４ビットの倍精度の場合では、（２）のように先頭１ビットが仮数部の符号（正または零：０、負：１）、指数部は７ビット、仮数部に５６ビットが割り当てられる。

（１）４バイト長単精度浮動小数点データ



仮数部符号　　▲
０：正、零　小数点
１：負

（２）８バイト長倍精度浮動小数点データ



仮数部符号　　▲
０：正、零　小数点
１：負

単精度浮動小数点では、指数部は７ビットで　$-2^{7-1} = -64 \leqq E \leqq 2^{7-1}-1 = +63$　まで表せるが、$-64 = 0$　となるように実際の指数部の値に６４を加えた数を指数部に格納する。

仮数部Ｍの範囲（絶対値）は　$0.100000 < M \leqq 0.FFFFFF$　となり、浮動小数点Ｎの範囲は　$0.1 \times 16^{-64} = 16^{-65} < |N| \leqq 0.FFFFFF \times 16^{63}$　となる。

１０進数に変換するため　$16^{-64} = 10^{-x}$，$16^{63} = 10^{y}$　とおくと、表現可能な数値範囲は　$10^{-78} \sim 10^{75}$　となる。また、有効桁数は、$2^{24} = 10^{7.2}$　であるので、１０進７桁である。

（３）仮数部の正規化

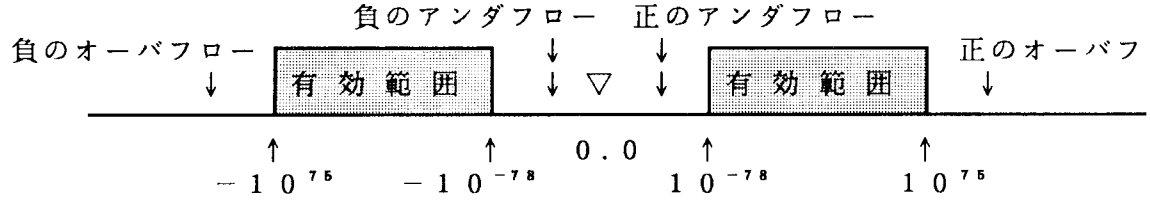
有効桁数を減らさないために、仮数の有効数字の前に４ビット以上０が並ぶ場合には、

・仮数を左へ４ビットシフトする
・指数部から１引く

を行う。これを「仮数部の正規化」という。

（４）オーバフロー、アンダフロー

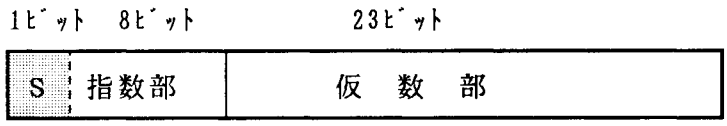
演算の結果が表現できる数値よりも大きい場合、オーバフロー（overflow）が生じたといい、演算結果が表現できる数値よりも０．０に近い場合、アンダフロー（underflow）が生じたという。単精度浮動小数点で表現できる値域を有効範囲として表すと、次のようになる。



（５）ＩＥＥＥ標準形式

浮動小数点のＩＥＥＥ標準形式を以下に示す。

（ａ）４バイト長単精度浮動小数点データ

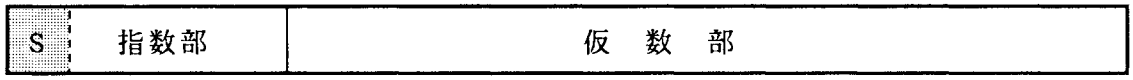


仮数部符号　　▲
　０：正、零　小数点
　１：負

指数部は１２７を加え非負の数とする。
指数部＝２５５，仮数部≠０のとき　数値を表現しないことを示す。
指数部＝２５５，仮数部＝０のとき　符号により正（負）の無限大を表す。
０＜指数部＜２５５のとき
　　（－１）^{s}２$^{e-127}$（１．ｆ）を表す。
　　ここで、ｓは符号、ｅは指数部、ｆは仮数部の値である。
指数部＝０，仮数部≠０のとき
　　（－１）^{s}２$^{-126}$（０．ｆ）を表す。
　　ここで、ｓは符号、ｅは指数部、ｆは仮数部の値である。

指数部＝０，仮数部＝０のとき　０を表す。

（ｂ）８バイト長倍精度浮動小数点データ



仮数部符号　　▲
　０：正、零　　小数点
　１：負

指数部は１０２３を加え非負の数とする。
指数部＝２０４７，仮数部≠０のとき　数値を表現しないことを示す。
指数部＝２０４７，仮数部＝０のとき　符号により正（負）の無限大を表す。
０＜指数部＜２０４７のとき
　　（－１）^{s}２$^{e-1023}$（１．ｆ）を表す。
　　ここで、ｓは符号、ｅは指数部、ｆは仮数部の値である。
指数部＝０，仮数部≠０のとき
　　（－１）^{s}２$^{-1022}$（０．ｆ）を表す。
　　ここで、ｓは符号、ｅは指数部、ｆは仮数部の値である。

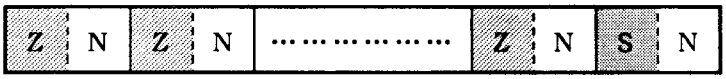
指数部＝０、仮数部＝０のとき、　０を表す。

１．５．３　ゾーン１０進数（アンパック形式）

ゾーン形式は、アンパック形式とも呼ばれ、１つの数値を表すのに１バイト（８ビット）を用いている。８ビットのうち、上位４ビットをゾーン部、下位４ビットを数値部と呼ぶ。１０進数の各桁をＢＣＤ（Binary Coded Decimal）コードを用いて表現し、１バイトに１０進数１桁が格納される

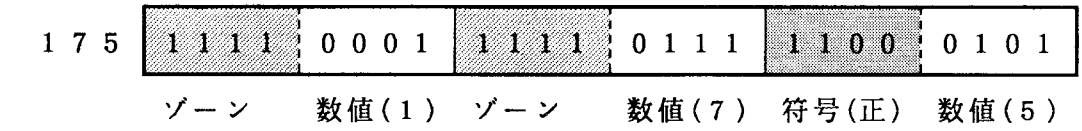
ゾーン部：データが数値である事を示す（１１１１）$_2$を格納する。ただし、最下位バイトのゾーンビットは、数値の正負を示す符号コード（正または零は１１００、負は１１０１）を格納する。

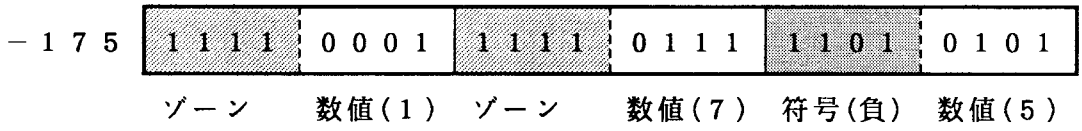
数値部：０～９の２進化１０進コードを格納する。



Ｚ：ゾーン部　　数値を表す（１１１１）$_2$
Ｎ：数値部　　０～９の２進化１０進コード
Ｓ：符号　　正負の符号を表す（正または零：１１００、または、１１１１）
（負：　　１１０１）

例）ゾーン形式で（１７５）$_{10}$、（－１７５）$_{10}$をＥＢＣＤＩＣコードで表すと、





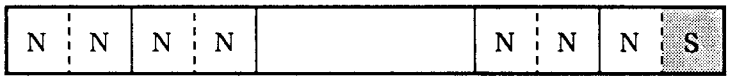
なお、ＡＳＣＩＩコードではゾーンを００１１で表す。

１．５．４　パック１０進数

パック形式は、１つの数値を表すのに４ビットを用いている。従って１バイトに２つの１０進数値を格納する事ができ、格納効率があがる。

数値部：１０進数の各桁を２進化１０進コード（ＢＣＤ）を用いて表現し、格納する。

符号　：最下位バイトの下位４ビットに正または零は（１１００）$_2$、
負は（１１０１）$_2$が格納される。



N：数値部　　０～９の２進化１０進コード
S：符号　　正負の符号を表す（正または零：１１００、負：１１０１）

例）パック形式で（３１４２７）$_{10}$、（－３１４２７）$_{10}$をＥＢＣＤＩＣコードで表すと、

３１４２７

００１１	０００１	０１００	００１０	０１１１	１１００
数値(３)	数値(１)	数値(４)	数値(２)	数値(７)	符号(正)

－３１４２７

００１１	０００１	０１００	００１０	０１１１	１１０１
数値(３)	数値(１)	数値(４)	数値(２)	数値(７)	符号(負)

また、（３１４２）$_{10}$のような偶数桁の場合は先頭に０があるものとして３バイトで表現する。

０３１４２

０００００	００１１	０００１	０１００	００１０	１１００
数値(０)	数値(３)	数値(１)	数値(４)	数値(２)	符号(正)

１．５．５　誤差

浮動小数点演算により生じる誤差について理解させ、誤差の発生を極力抑える演算の方法を理解させる。

（１）情報落ち

絶対値の大きな数と絶対値の小さな数との間の加減算を行う場合に生じる現象である。
例えば、

１００００＋１＝１００００１

であるが、有効桁数が５桁の場合は１の位は無視されて１００００となる。

また、３つの数を有効桁数５桁で加算する場合、大きい順から加算するときと、小さい順から加算するときでは、計算結果が異なる。

① 大きい順から加算する
（１００００＋９）＋１
１００００＋９≒１００００
１００００＋１≒１００００

② 小さい順に加算する
１００００＋（９＋１）
９＋１＝１０
１０＋１００００＝１０００１０

（２）けた落ち

ほぼ等しい数値の減算を行うと、有効数字の桁数が少なくなり、精度が低下する。

たとえば、

１２３４５６－１２３４５５＝１

となり、有効数字の桁数は６桁から１桁に落ちてしまう。

浮動小数点で計算しても同様で、
１５５の平方根と１５４の平方根を引き算することを考える。
有効桁数８で考えると、

１５５の平方根は１２．４４９８９９
１５４の平方根は１２．４０９６７３　であるので、差をとると

０．０４０２２６　となり有効桁が５桁に減ってしまう。

この例では、「分子の有理化」を行い

$$\mathrm{sqrt}(155)-\mathrm{sqrt}(154) = \frac{155-154}{\mathrm{sqrt}(155)+\mathrm{sqrt}(154)}$$

$$= \frac{1}{24.859572} = 0.040225957$$

が得られ、有効桁数が保たれて、桁落ちが防げる。

指導上の留意点

ポイント

① 各々の数値表現方法の特徴・違いを理解させる。

② 固定小数点で負数を表す各種方法の違いを理解させる。

③ 補数の求め方、補数による演算も理解させる。

④ 浮動小数点による表現方法、表す数の範囲を理解させる。

⑤ 浮動小数点のオーバフロー、アンダフローを理解させ、値の有効範囲を教える。

⑥ １０進数による表現方法（ゾーン１０進数、パック１０進数）を理解させる。

⑦ 誤差について関心をもたせ、数値演算を行うときには常に有効桁に気を使わせる。

⑧ 「情報落ち」「けた落ち」について理解させる。

用語

固定小数点　補数　１の補数　２の補数　浮動小数点　指数部　仮数部
倍精度浮動小数点　有効桁数　正規化　オーバフロー　アンダフロー　ゾーン１０進数
パック１０進数　情報落ち　けた落ち

講師ノート

第２種情報処理技術者試験

・数の表現に関する問題はしばしば出題されるので充分に理解しておく必要がある。

・誤差については、試験対策としてのみならず、数値計算を行う場合には「有効桁数」を常に意識しておく事が重要であるので、ここで充分に理解させる。