

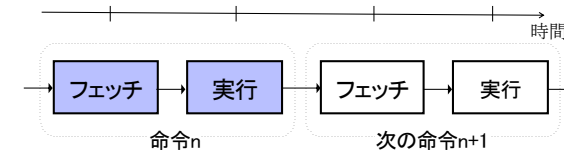
計算機方式論

第1章 計算機の基本構成 #2

1

命令の実行順制御-逐次制御

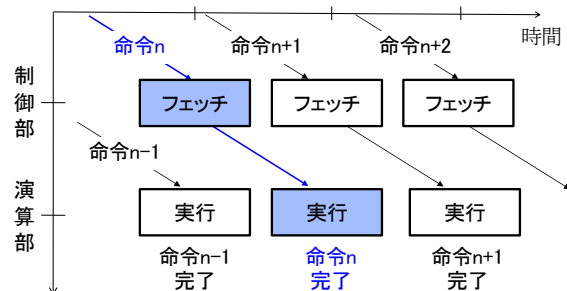
- 命令のフェッチサイクル後に、実行サイクルを行い、その後、次の命令のフェッチ・実行サイクルを行っていく。



2

命令の実行順制御-先回り制御

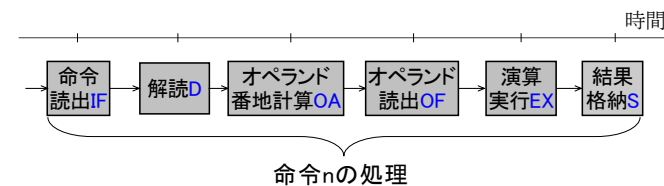
- フェッチサイクルと実行サイクルとを同時実行
- CPUの制御部と演算部とが独立して動作できることで実現



3

ステージ分割

- フェッチ・実行サイクルをステージに分割
- 各ステージが独立して動作できるようにする
- 演算命令は、6ステージに分割される



4

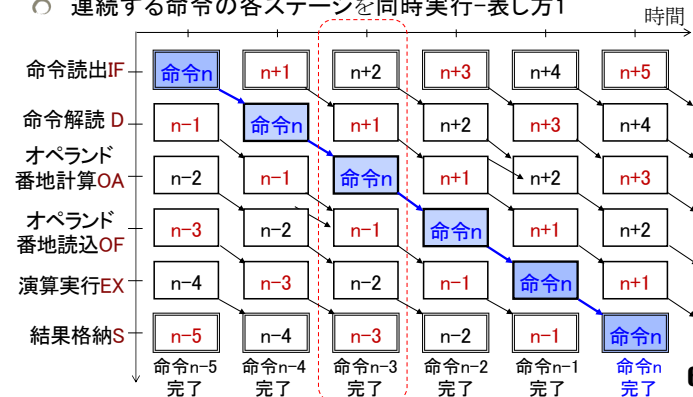
各ステージの動作

- **命令読み出しIF**
PCの指す命令をIRにフェッチ。PCを1増す。
- **解読D**
IR中の命令を解読し、制御系列を選択。
- **オペランド番地計算OA**
オペランドが主記憶のとき、**主記憶番地**を計算し、MARに。
 $\text{メモリ減算 } S, 2, 512(3) \rightarrow (GR2) - (512 + (GR3)) \rightarrow GR2$
- **オペランド読み出しOF**
主記憶番地の内容を、MDRに読み出し、ALUに送る。
オペランドがレジスタのとき、このステージで、ALUに送る。
- **演算実行EX**
ALUを用い、演算を施す。
- **結果格納S**
演算結果をレジスタや主記憶に保存する。
分岐命令では、このステージで、分岐番地をPCにセット。

5

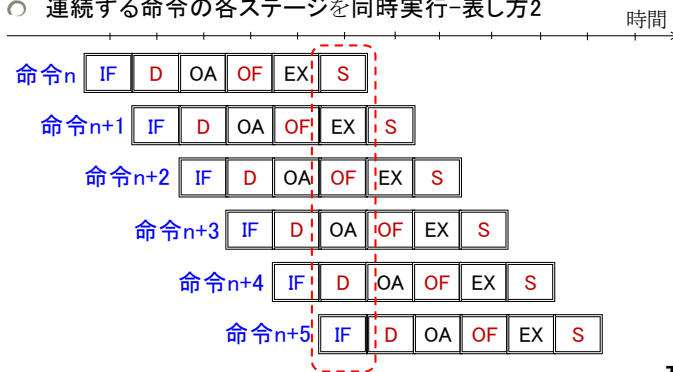
命令の実行順制御-パイプライン制御

- 連続する命令の各ステージを同時実行-表し方1



命令の実行順制御-パイプライン制御

- 連続する命令の各ステージを同時実行-表し方2



パイプライン(先回り)制御の特長

- **6つのステージ**(2つのサイクル)を同時に動作
⇒ 単位時間にひとつの命令相当分の処理
- 理想的には、逐次制御の**6倍**(2倍)のスループットが得られる

8

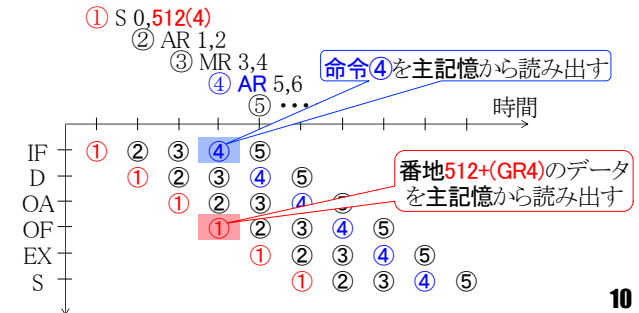
パイプライン制御の問題点

- すぐ前の命令の結果を次の命令で使う
- 分岐命令
↓
- 阻害要因 (ハザード, hazard) となり、制御が立ち往生する (インタロック, interlock)
- 構造ハザード
データハザード
制御ハザード

9

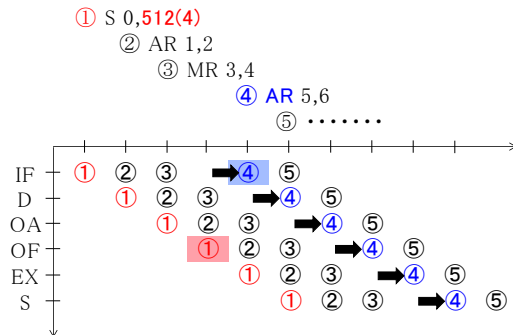
構造ハザード

- CPUの内部構成が原因。資源の競合で起きる
- 主記憶アクセスの競合。
①の512+G4番地のデータ読出と④の命令読出とが競合



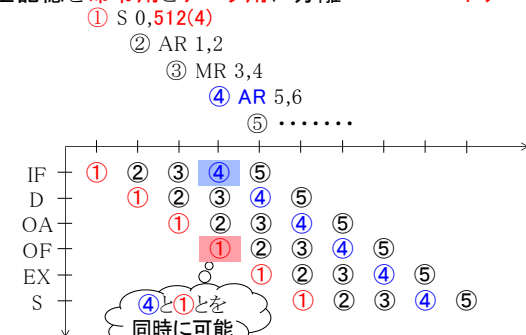
構造ハザード: 主記憶

- ①の512+G4番地のデータ読出と④の命令読出とが競合
下図のように、競合ステージ以降を1クロック遅らす



構造ハザード-主記憶 解決

- 資源の多重化で解決:
主記憶を命令用とデータ用に分離... ハーバードアーキテクチャ



構造ハザード-ALU

- ALU使用でのOAとEXとの競合:

ALUを①の実行ステージと③のオペランド番地計算で同時使用

① MR 1,2

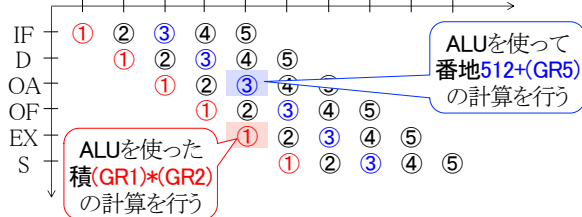
② AR 3,4

③ S 5,512(5)

④ S 0,2

⑤

ALU使用でのOAとE、
バス使用やIR使用等
での競合もある



13

データハザード

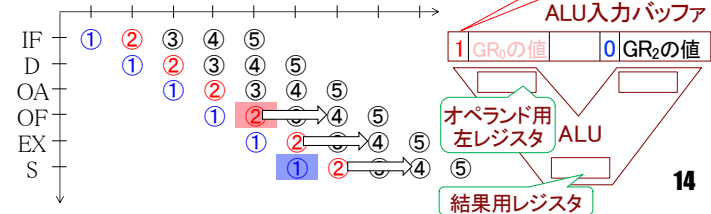
- データの依存関係に基づく。

次の命令が前の命令の結果を使う(フロー依存)

① AR 0,1 加算結果はGR0に格納

② MR 0,2 GR0とGR2とのかけ算

左被演算子が
レジスタ1のため
ALUが値待ち



14

データハザード 解決

- 依存関係にある命令をできるだけ離して解決。
命令の位置を最適化する命令スケジューリングを

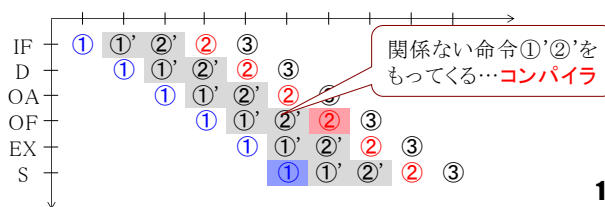
コンパイラで行う・・・ソフトウェアで解決

① AR 0,1 加算結果はGR0に格納

①' 関係ない命令を移動してくる

②' 関係ない命令を移動してくる

② MR 0,2 GR0とGR2とのかけ算



15

制御ハザード

- 分岐命令と分岐先命令とは制御依存関係にある。
これに基づくハザード。

②～⑥のステージがむだになる。

① BM L 演算結果が負ならば、L番地へ分岐

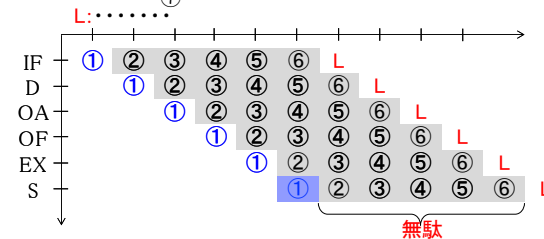
②

③

④

⑤

⑥



16

制御ハザード 解決①

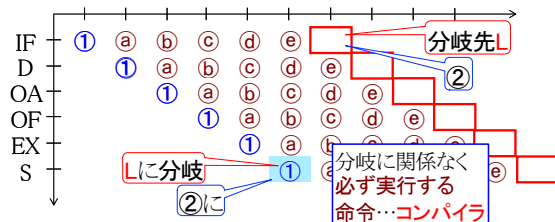
- 分岐命令の次の番地以降(分岐遅延スロット)には、分岐に関係なく**必ず実行する命令**を置く
… ソフトウェアで解決

① BM L 5ステージ後に、条件分岐を実行

② a~e: 分岐に関係なく**必ず実行する命令**

③ ……

L: ……



17

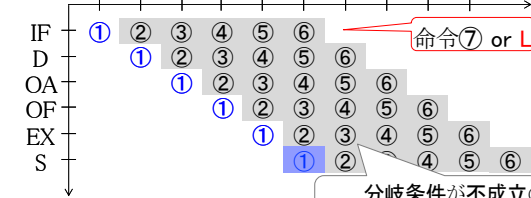
制御ハザード 解決②

- 分岐予測を行う。次の番地以降に**分岐確率の高い命令**を置く。「直前の演算**結果が正**」の確率が高いとする。

① BM L 「結果が**負**」のとき、分岐してL以降の命令を実行

②~⑥: 「結果が**正**」のとき実行する命令。**分岐確率の高い命令**

⑦



分岐条件が不成立のとき

①の時点の環境に戻す機構が必要

- 分岐予測を動的に行う方式もあるが、複雑な機構を要する。

18