

Web インタフェースによる学生実習向け遠隔プログラミング環境の整備 Remote Programming Environment with Web User Interface for Students Training

神原 弘之[†]
Hiroyuki Kanbara

大迫 裕樹^{‡§}
Yuuki Oosako

國枝 義敏[§]
Yoshitoshi Kunieda

1 はじめに

教育機関では、新型コロナウイルス感染症の拡大に伴い、講義のオンライン受講に加え、実験や実習を在宅で実施する必要性が高まっている。

計算機アーキテクチャ入門実験あるいは組込みシステムのプログラミング実習には、C コンパイラやアセンブラが動作する開発環境を用いて、市販 CPU を用いたマイコンボードをプログラムするカリキュラムが広く採用されている。大学や高専の実習室には、開発環境を利用するための PC とマイコンボードの実機があり、CPU のプログラミングとその動作確認を行うことができる。

しかし在宅で実習を行う場合「手元の PC で開発環境が動作するか」あるいは「自宅に送付されたマイコンボードの使用方法を遠隔からどのようにサポートするか」といった問題が生じる。このため実習室と同等のカリキュラムを、在宅でそのまま実施することは難しい。

マイコンボードの実機の代わりに、PC 上で動作する命令セット・シミュレータを用いて、CPU のプログラミングと動作確認を仮想的に実施する実習も広く行われている。しかし、命令セット・シミュレータのソフトウェアが、学習者の自宅の PC で動作するかどうか、課題となっている。

本稿では、KUECHIP-3F と呼ばれる教育向けに設計された組込みプロセッサについて、アセンブルと命令シミュレーションを Web ブラウザ上で実行できるプログラミング環境を整備し、実機を用いて行ってきた学生実習を、遠隔で実施した事例について報告する。

2 教育用プロセッサ KUECHIP-3F について

2.1 概要

KUECHIP-3F は、計算機の動作原理の理解と、FPGA 設計教育の題材という目的に特化して設計された、16bit アキュムレータ方式のプロセッサである [1]。

KUECHIP-3F を用いた教育用計算機のブロックダイ

アグラムを図 1 に示す。点線内が FPGA 内に実装されている部分である。入力データバス (DBi)、出力データバス (DBo)、アドレスバス (AB) の他に、内部状態を観測するための観測バス (OB) があり、各々は 16bit 幅である。観測バス (OB) がレジスタやメモリの値を外部に出力することにより、モニタープログラムを介さずに、プロセッサの内部動作を観察することができる。また、DBi バス経由で、内部のレジスタやメモリの内容を書き換えることができる。

演算用レジスタにはアキュムレータ (ACC)、インデックスレジスタ (IX)、スタックポインタ (SP) があり、演算器 (ALU) を 1 個備えている。ALU による演算の結果により、零フラグ (ZF)、ネガティブフラグ (NF)、桁あふれフラグ (VF)、桁上げフラグ (CF) の 4 つのフラグ (FLAG) の値が更新される。命令の実行制御用レジスタには、プログラムカウンタ (PC)、命令レジスタ (IR) があり、メモリアクセスの際のアドレスを保持するメモリアドレスレジスタ (MAR) がある。

2.2 命令セット

表 1 に KUECHIP-3F の命令セットを示す。算術論理演算、シフト演算、分岐、ロード/ストア、実行制御、スタックポインタ操作の計 46 種の固定長の命令をサポートしている。メモリアクセスは、即値アドレス、直接アドレス、インデックスレジスタによる修飾アドレス、スタックポインタによる修飾アドレスの 4 つのアドレス指定方法に対応している。

KUECHIP-3F の命令実行フェーズを表 2 に示す。P0、P1 は命令取り出しフェーズであり、P2、P3、P4 が命令の実行フェーズである。1 命令は最大 5 クロックフェーズで実行される。

2.3 プログラムの実行モード

KUECHIP-3F は、個々の命令の動作そしてプロセッサ内部での命令の解釈と実行の仕組みを理解するため

- HaLT 命令が実行されるまで命令が連続して実行され、作成したプログラムの正しさを確認する通常動作 (SS) モード
- 1 命令の実行ごとに動作を止め、命令セットレベルシミュレータのように実行結果を確認できるシングルインストラクション (SI) モード、

[†] 京都高度技術研究所, ASTEM RI

[‡] 関西学院大学, Kwansai Gakuin University, 現在はグリーン株式会社

[§] 立命館大学, Ritsumeikan University

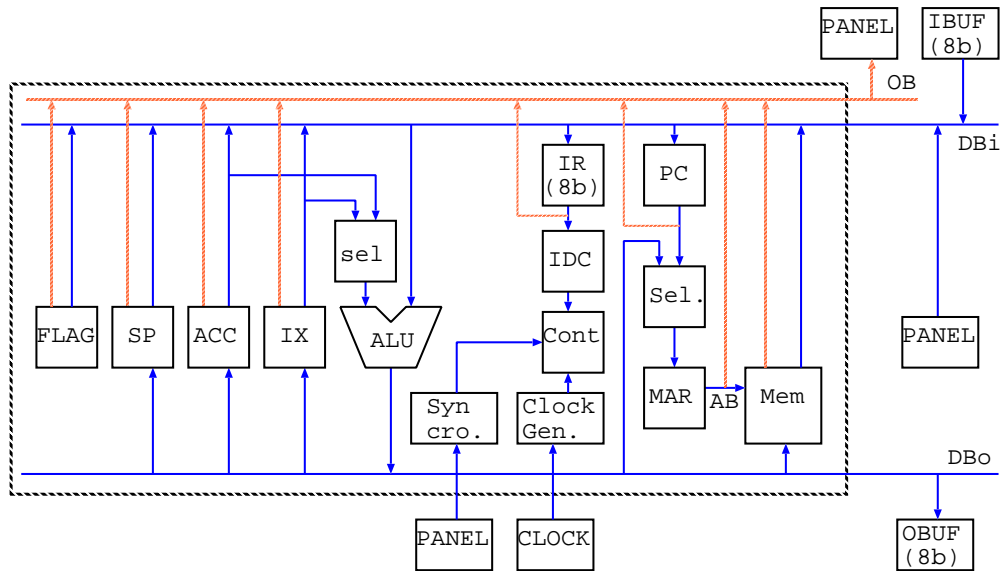


図1 KUECHIP-3F のブロックダイアグラム

- 命令の取り出し／デコード／ALU 演算／メモリへのロードストアがどのように行われるかを 1 フェーズごとに観察するためのシングルフェーズ (SP) モード、

の 3 つの実行モードから 1 つを選択して機械語プログラムを実行することができる。

2.4 設計と実装

KUECHIP-3F プロセッサは VHDL 言語を用いて設計し、Xilinx 社の Artix-7 FPGA を用いて実装された。65536 ワード × 16bit の内蔵メモリ (Mem) は FPGA のブロック RAM 機能を用いて実現した。

2.5 KUECHIP-3F を搭載した教育用ボード

KUECHIP-3F をプログラミングし、その内部の命令の解釈実行を観測することができる、学生実習向け教育用ボードを開発した。本教育用ボードの概観を図 2 に示す。KUECHIP-3F プロセッサは、本ボード上の Digilent Cmod A7 FPGA モジュールに実装される。

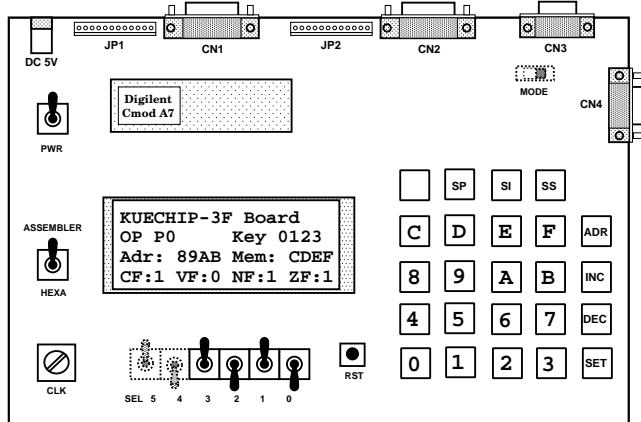


図2 KUECHIP-3F を搭載した教育用ボード (KR-CHIP)

教育用ボードは 20 文字の英数字を 4 行表示する蛍光表示管デバイスを搭載している。この表示管には

- メモリのアドレスとその値 (命令/データ)
- KUECHIP-3F プロセッサ内部のレジスタの値
- 16 進キーボードで入力した 16bit の値
- 現在の実行状態とクロックフェーズ

が表示される。16 進数による値の表示に加え、逆アセンブルにより KUECHIP-3F のアセンブリ言語で命令を表示することも選択できる。

この教育用ボードは KUECHIP-3F プロセッサの他に、

- KUECHIP-3F の前身となった 8bit アキュムレータ方式の KUE-CHIP2 プロセッサ [2][3][4][5]、
- 32bit 5 段パイプラインの RUECHIP2 プロセッサ [6][7]

の 3 つの教育用途に特化して設計されたプロセッサを搭載している [8]。

2020 年度の前期は、これまで実習室で本教育用ボードを用いて行われてきた学生実験を、学生は在宅のままで実施することになった。このため、次章で紹介する Web インタフェースによる命令セット・シミュレータとアセンブラを整備し、学生実習のカリキュラムを遠隔で実施した。

3 KUECHIP-3F の命令セット・シミュレータとアセンブラ

遠隔での学生実習を行うにあたり、ユーザのプラットフォームに依存せず、web ブラウザのみで

- KUECHIP-3F CPU のアセンブリ記述プログラムのアセンブル
- 命令セットレベルのシミュレーション

を行うことができる環境を作成した。

3.1 web アセンブラ/シミュレータの仕組み

アセンブラ・シミュレータの動作の概観を図 3 に示す。

表1 KUECHIP-3F の命令セット

算術論理演算		シフト演算	
ADD	ADD	SRA	Shift Right Arithmetically
ADC	ADd with Carry	SLA	Shift Left Arithmetically
SUB	SUBtract	SRL	Shift Right Logically
SBC	SuBtract with Carry	SLL	Shift Left Logically
CMP	CoMPare	RRA	Rotate Right Arithmetically
AND	AND	RLA	Rotate Left Arithmetically
OR	OR	RRL	Rotate Right Logically
EOR	Exclusive OR	RLL	Rotate Left Logically
分岐		ロード/ストア	
BA	Branch Always	LD	LoaD
BVF	Branch on oVerFlow	ST	STore
BNZ	Branch on Not Zero	実行制御	
BZ	Branch on Zero	HLT	HaLT
BZP	Branch on Zero or Positive	NOP	No OPeration
BN	Branch on Negative	IN	INput
BP	Branch on Positive	OUT	OUTput
BZN	Branch on Zero or Negative	SCF	Set Carry Flag
BNI	Branch on No Input	RCF	Reset Carry Flag
BNO	Branch on No Output	スタックポインタ操作	
BNC	Branch on No Carry	INC	INCrement
BC	Branch on Carry	DEC	DECrement
BGE	Branch on Greater than or Equal	PSH	PuSH
BLT	Branch on Less Than	POP	POP
BGT	Branch on Greater Than	CAL	CALl
BLE	Branch on Less than or Equal	RET	RETurn

アセンブラ (JavaScript) および命令セットシミュレータ (Processing) は web サーバから静的コンテンツとしてサブされ、フロントエンド側で完結した SPA (single page application) として動作する。Processing ファイルは processing.js^{*1}により、解釈・実行される。

図中の web-ui.js はユーザの操作に合わせて適切な処理を行い、画面表示の更新を行う等、UI (user interface) まわりの機能を提供する。例えば、ユーザがアセンブルの操作を行った場合、web-ui.js は assembler.js にアセンブリプログラムを渡し、結果として機械語プログラムを受け取る。シミュレーションを行う場合は、web-ui.js から Processing 側に状態 (レジスタやメモリの値) を渡し、次の状態を受け取る。この時、ユーザ操作に合わせてフェーズ・命令単位でシミュレーションが進行する。なお、processing.js 内のデータに直接アクセスすることはできないが、Processing 側のコードで必要な getter/setter を作成しておくことにより、Processing.getInstanceById で得られる Processing オブジェクト内に getter/setter が作られ、それらを介してデータの受け渡しを行うことができる。

3.2 アセンブラの実装

KUECHIP-3F 用のアセンブラを TypeScript で実装した。本アセンブラは、2.2 節の命令を用いたアセンブリプ

ログラムを入力とし、機械語プログラムを生成する。生成された機械語プログラムは KUECHIP-3F ボードおよび、後述する命令セットシミュレータ上で動作する。

本アセンブラは node およびブラウザ向けの JavaScript にトランスパイルし、ターミナルおよびブラウザ上で動作させることができる。web 版は命令セットシミュレータに同梱しているため、アセンブラが生成した機械語プログラムをそのままシミュレートすることができる。

3.3 命令セットシミュレータの実装

命令セットシミュレータは、ビジュアルなシミュレーション結果の表示を目指して Processing 言語で実装された [1] を元に、機能および UI の改良を行い、3.2 節のアセンブラを統合した。

本シミュレータは、KUECHIP-3F CPU の特徴である、フェーズ単位での命令解釈・実行におけるレジスタ転送レベルの挙動の観測が行うことができる。また、ユーザは状況に応じて任意の実行速度でシミュレータを動作させることが可能である。

3.4 web シミュレータ・アセンブラの UI

命令セットシミュレータおよび統合されたアセンブラの画面をそれぞれ図 4, 5 に示す。いずれも、最低限の UI パーツのみで構成し、状況に応じてコンポーネントの見た目を変えることで直感的な操作が可能になるよう配慮した。

*1 <http://processingjs.org>

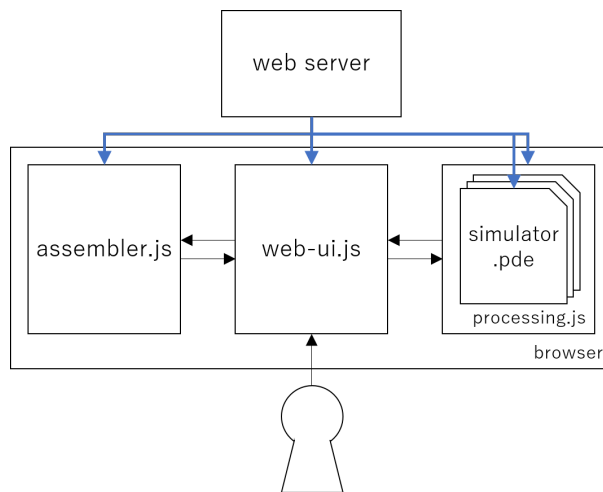


図3 アセンブラ・シミュレータの動作

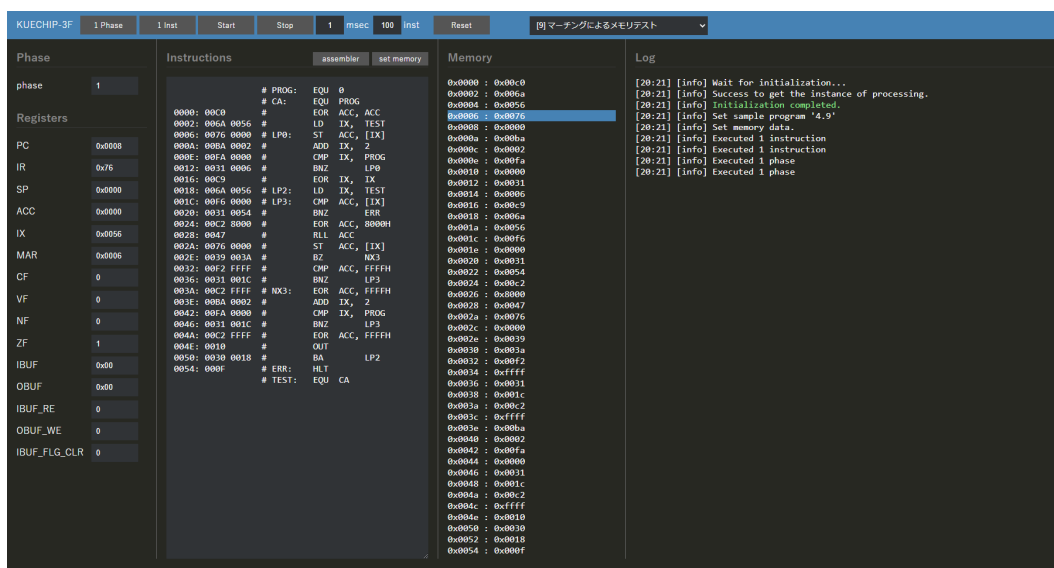


図4 命令セットシミュレータの画面

3.5 実装についての考察

3.5.1 Processing と TypeScript

命令セットシミュレータ [1] で用いている Processing と、アセンブラで用いている TypeScript それぞれの長所・短所について考察する。上記シミュレータは JVM 上でのローカル実行も可能だが、以下では web 上で SPA として動作させることを前提とする。

Processing は、開発時に processing.js と併用して本来の処理系を使用することで、静的型付き言語として用いることができ、また、グラフィカルな描画がカジュアルにできることが利点である。一方、欠点としては、processing.js が本来の処理系とは異なる処理系であるため、web 上でのみ問題が発生するケースが存在することが挙げられる。この場合、processing.js を経由して動作するコードのデバッグはやや手間がかかる。また、ユーザからのテキスト入力が必要な場合は、Processing のみで UI を完結させることができず、別途フォームを作成し、UI 制御用の JavaScript プログラムと Processing のプログラム間でデータの受け

渡しを行う必要がある。

TypeScript は上記の静的型付け言語としての利点を持ちつつ、トランスパイルすることでブラウザの JavaScript エンジンでそのまま動作するため、Processing の場合と比べて、フォームとの連携やデバッグ等が容易である。また、CLI (command line interface) では Node.js 上で実行できるためロジック部分のテストが容易であり、CI/CD (continuous integration/continuous delivery) との相性も良いと言える。

3.5.2 CGI 実装との比較

KUE-CHIP2 ではアセンブラ/シミュレータを CGI で提供している。^{*2} CGI の場合、アセンブラ/シミュレータをサーバサイドで用意し、ユーザの操作毎にリクエストを行うことになる。この時ブラウザのページ遷移が生じるため、実現できる UX (user experience) に制限が生じる。

一方、処理をフロントエンドに寄せ、SPA として実装す

^{*2} <http://www.metsa.astem.or.jp/kue2/kue-chip2/>

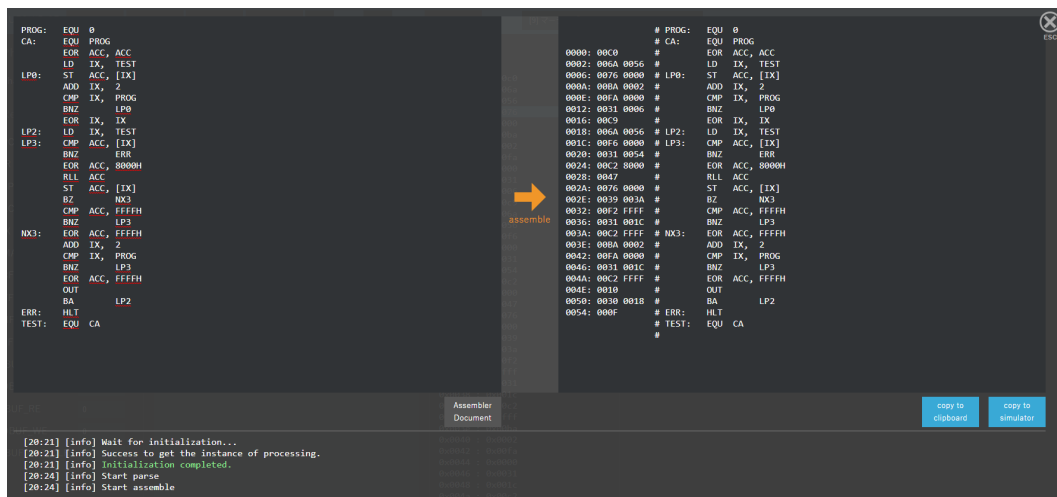


図5 アセンブラの画面

ることにより、柔軟な UI/UX を提供することが可能になる。本システムの例では、UI コンポーネントの見た目の変化を目に付きやすくすることにより次の操作を自然に誘導する、シミュレーション中は状態を表示している DOM 要素のみを更新し、実行過程を観察しやすくする等の工夫を行っている。

4 オンラインでの学生実験の実施

現在の一般的なコンピュータのアーキテクチャとその動作を詳細に理解することを目標に、2018 年度より立命館大学情報理工学部セキュリティ・ネットワークコースでは、学部 2 回生の前期に KUECHIP-3F プロセッサを搭載した教育用ボードを用いた実験科目を配置している。具体的には 2 コマ× 7 週間で“機械語プログラミングを実体験する”、“命令の実行過程をつぶさに観察する”課題を用意し学生実験を実施している。

2020 年度は、新型コロナウイルス感染症への対策のため、実験室で実際に実機を利用する形態をあきらめざるを得ず、やむなく 3 章で既述のアセンブラおよびシミュレータのみを利用するオンライン形態で特例的に実施してきている。正規実験時間中は、教員およびティーチングアシスタント 8 名が、ZOOM、特にそのブレイクアウトセッション機能を利用し受講生個別対応することで可能な限りきめ細かな質問対応も実施している。操作性の面では、実機に及ばないようではあるが、シミュレータが実機そのままに動作するため、これまでの様子を見る限り、受講生は上記の教育目標に到達できているように見受けられる。

5 おわりに

本稿では、Web ブラウザ上で動作するプロセッサの命令セット・シミュレータとアセンブラを整備し、これまでプロセッサを搭載した教育用ボードの実機を用いて実施してきた学生実習を、遠隔で実施した事例を紹介した。今回整備した命令セット・シミュレータとアセンブラは、Google Chrome, Mozilla Firefox (Version 78 以降) で

動作することを確認した。PC の OS の種類 (Windows, macOS, Linux) を問わず、アセンブリ記述プログラムのアSEMBルと命令実行のシミュレーションを、学習者は在宅で行うことが可能となった。さらに iPad OS の Safari ブラウザでも動作しており、タブレット端末での遠隔学習も可能であると考えられる。今後の課題としては、命令シミュレーションを中心とした在宅での予習/復習と、実習室の実機を使用した学生実験の組み合わせによる、より効果的な学生実習のカリキュラムのあり方があげられる。

謝辞

本稿の開発を行なうにあたり、有益なアドバイスを頂いた元立命館大学の中谷嵩之氏、元京都大学の矢野正治氏、元関西学院大学の田村真平氏に感謝致します。

参考文献

- [1] 東香実, 大迫裕樹, 柴田敦也, 神原弘之, 國枝義敏: “教育用 16bit プロセッサ KUECHIP-3F の開発,” 2017 年度 情報処理学会関西支部 支部大会, A-04 (Sept. 2017).
- [2] 越智裕之, 澤田宏, 岡田和久, 上嶋明, 神原弘之, 濱口清治, 安浦寛人: “計算機工学・集積回路工学教育用 マイクロプロセッサ KUE-CHIP2,” 情報処理学会研究報告, 1992-ARC-96-13 (1992).
- [3] 神原弘之, 越智裕之, 澤田宏, 濱口清治, 岡田和久, 上嶋明, 安浦寛人: “KUE-CHIP2 教育用ボードリファレンスマニュアル,” 京都高度技術研究所 (1992).
- [4] 神原弘之, 越智裕之, 澤田宏, 濱口清治, 岡田和久, 上嶋明, 安浦寛人: “KUE-CHIP2 設計ドキュメント,” 京都高度技術研究所 (1992).
- [5] 川端英之, 弘中哲夫, 寺内衛: “情報工学基礎実験,” 大学教育出版 (Arp. 2005).
- [6] 神原弘之, 金城良太, 戸田勇希, 矢野正治, 小柳滋: “パイプラインプロセッサを理解するための教材 RUECHIP1,” 2009 年度 情報処理学会関西支部 支部大会, A-09 (2009).
- [7] デイビット A パターソン, ジョン L ヘネシー: “コンピュータの構成と設計 第 5 版,” 日経 BP 社 (2014).
- [8] 柴田敦也, 大迫裕樹, 東香実, 中野和香子, 神原弘之: “8, 16, 32 ビットのプロセッサを搭載した教育用計算機: KR-CHIP,” 2018 年度 情報処理学会関西支部 支部大会, A-101 (Sept. 2018).

表 2 KUECHIP-3F の各命令の実行フェーズ表

Instruction	Phase	P0	P1	P2	P3	P4
HLT		(PC) → MAR (PC+2) → PC	(Mem) → IR	HLT		
LD	IX, SP			SP → IX		
LD	SP, d			(PC) → MAR (PC+2) → PC	(Mem) → SP	
INC				+2 → (SP) → ALU → SP		
DEC				-2 → (SP) → ALU → SP		
ADD	SP, d			(PC) → MAR (PC+2) → PC	(SP) → (Mem) → ALU → SP	
SUB	SP, d			(SP-2) → MAR (SP-2) → SP	(ACC(IX)) → Mem	
PSH	ACC (IX)			(SP) → MAR (SP+2) → SP	Mem → ACC(IX)	
POP	ACC (IX)			(SP-2) → MAR (SP-2) → SP	(PC+2) → Mem (PC) → MAR	(Mem) → PC
CAL				(SP) → MAR (SP+2) → SP	(Mem) → PC	
RET				No Operation		
NOP				(ACC) → OBUF		
OUT					0 → OBUF_WE	
IN				(IBUF) → ACC 0 → IBUF_RE	0 → IBUF_FLG_CLR	
Bcc				(PC) → MAR (PC+2) → PC	STATUS CHECK (Mem) → PC (if condition)	
Ssm				TCF SET	NF, ZF, VF, CF Set	
Rsm				SHIFT A		
LD	ACC IX d [d] [IX+d] [SP+d]			(A) → B		
				(PC) → MAR (PC+2) → PC	(Mem) → A (Mem) → MAR (IX(SP)) → (Mem) → ALU → MAR	(MEM) → A
ST	[d] [IX+d] [SP+d]			(PC) → MAR (PC+2) → PC	(Mem) → MAR (IX(SP)) → (Mem) → ALU → MAR	(A) → (Mem)
SBC	ACC	(A) → (B) → ALU → A				
ADC	IX	[(CF)] → NF, ZF, VF [, CF] SET				
SUB			(A) → (B) → ALU → A			
ADD	d		[(CF)] → NF, ZF, VF [, CF] SET			
EOR		(PC) → MAR (PC+2) → PC	(Mem) → MAR	(A) → (B) → ALU → A		
OR	[d]		(IX(SP)) → (Mem) → ALU → MAR	[(CF)] → NF, ZF, VF [, CF] SET		
AND						
CMP	[IX+d] [SP+d]					