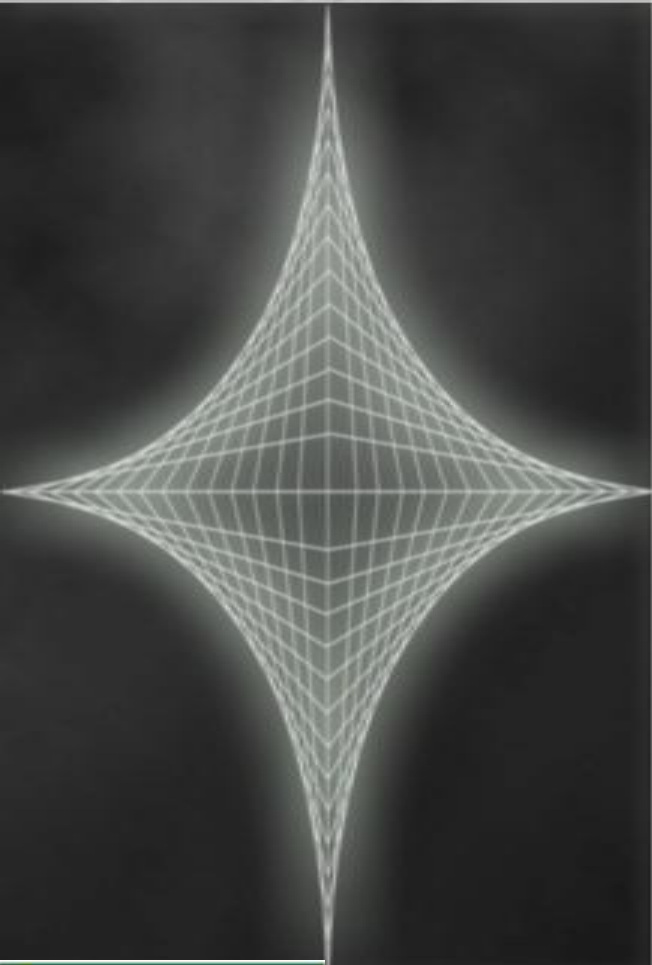


Structured Analysis & Structured Design <CFG Generator>



Class B – Team 1

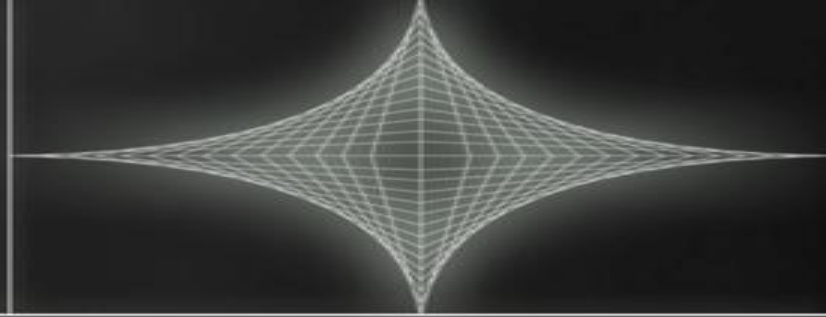
200611523 한정석

200611525 홍준택

200611230 육근웅

200711445 엄호경

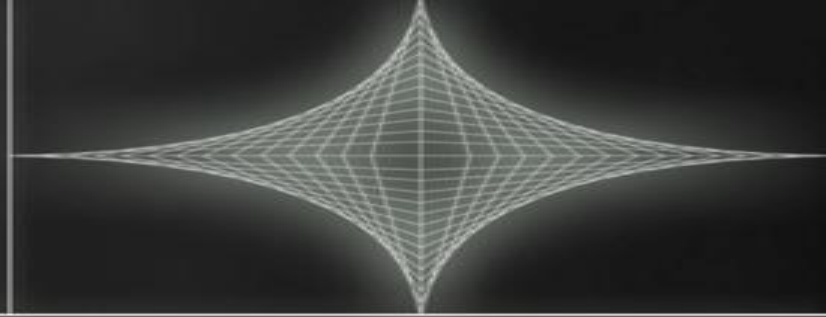
References



- Modern Structured Analysis, Edward Yourdon, 1989.
- Introduction to System Analysis and Design: a Structured Approach, Penny A. Kendall, 1996.
- Zhou Qun, Kendra Hamilton, and Ibrahim Jadalowen (2002). Structured Analysis and Structured Design (SASD) – Class Presentaion
<http://pages.cpsc.ucalgary.ca/~jadalow/seng613/Group/>



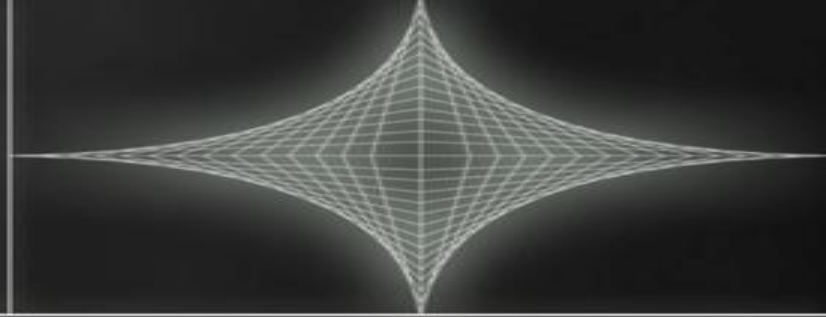
Contents



- Structured Analysis
 - Statement of Purpose
 - System Context file Diagram
 - Event List
 - DFD(Data Flow Diagram)
 - Data Dictionary
 - Process Specification
- Structured Design
 - Structured Charts – Transform Analysis
 - Structured Charts(Basic)
 - Structured Charts(Advanced)



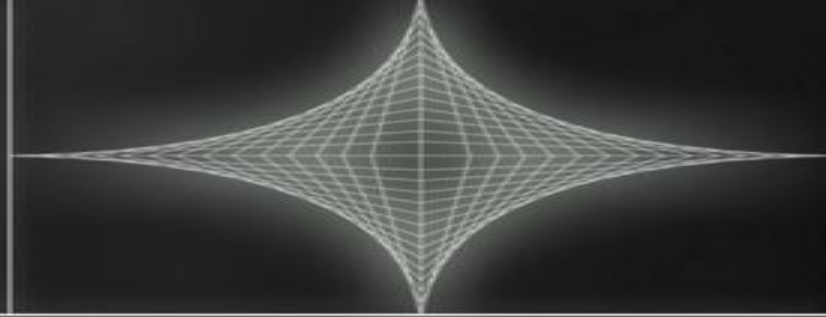
Statement of Purpose



- .c 파일을 입력 받아서 CFG를 출력시키는 프로그램이다.
- 입력받는 파일은 c 언어로 작성된 .C 파일만 허용하며 이외에는 도움말을 출력한다.
- 잘못된 형태의 명령어를 입력하면 도움말을 출력한다.
- User Interface는 CUI형태로 제공되며 Command Line 입력 방식으로 수행된다.
- 명령어 입력(./gc inputcode.c result.txt) argv[]는 2개를 받는다. 이외의 입력은 오류로 처리하게 된다.
- 단일 파일로 되어 있는 코드에 대하여 작동한다.
- CFG 생성시 주석은 제외한다.
- 변환을 시작하기 전에 변환의 시작을 사용자에게 알린다.



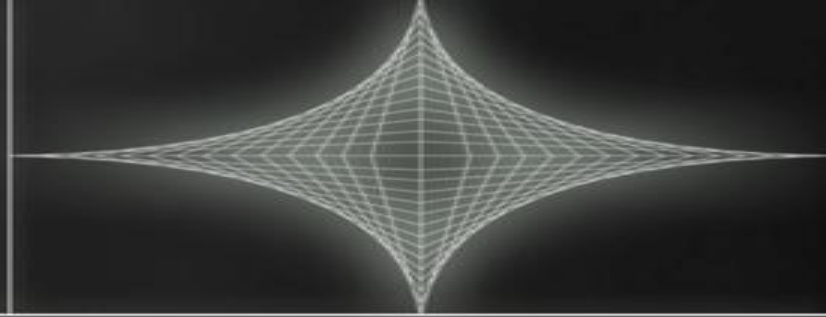
Statement of Purpose



- .c 파일을 입력 받아서 CFG를 출력시키는 프로그램이다.
- 입력받는 파일은 c 언어로 작성된 .C 파일만 허용하며 이외에는 도움말을 출력한다.
- 잘못된 형태의 명령어를 입력하면 도움말을 출력한다.
- User Interface는 CUI형태로 제공되며 Command Line 입력 방식으로 수행된다.
- 명령어 입력(./gc inputcode.c result.txt) argv[]는 2개를 받는다. 이외의 입력은 오류로 처리하게 된다.
- 단일 파일로 되어 있는 코드에 대하여 작동한다.
- CFG 생성시 주석은 제외한다.
- 변환을 시작하기 전에 변환의 시작을 사용자에게 알린다.



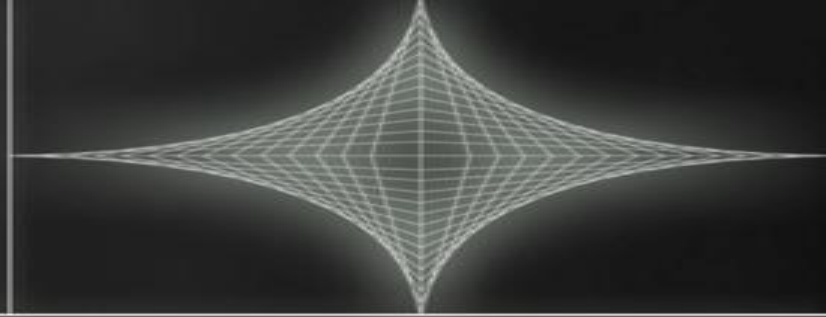
Statement of Purpose



- ▶ 화면에 출력할 Control Flow Graph는 Node \rightarrow Node 형식으로 표현하며 \rightarrow 는 일반적인 Edge, \Rightarrow 는 조건분기문에서의 Edge, \gg 는 Loop문에서의 Back Edge를 나타내기로 한다.



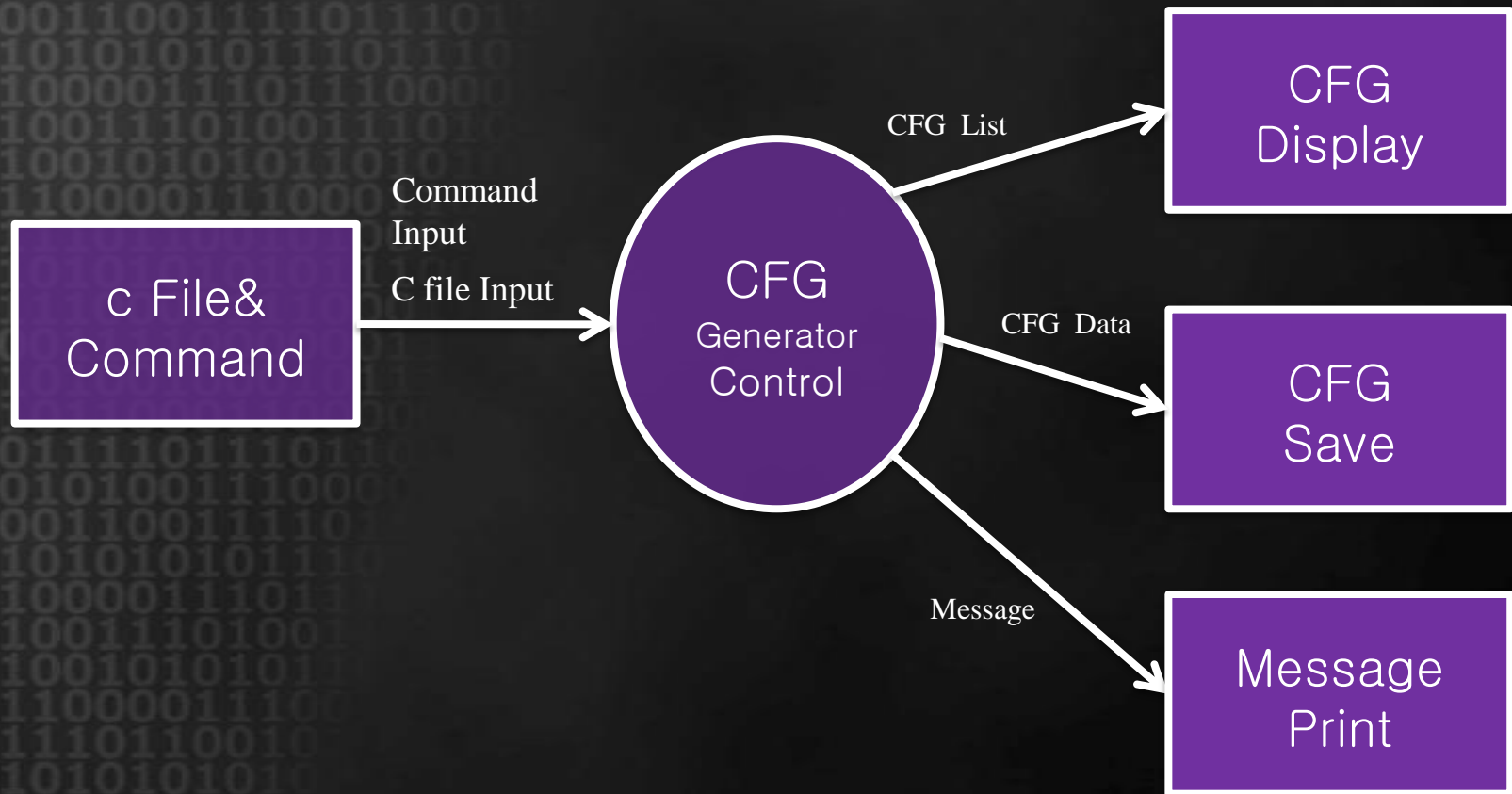
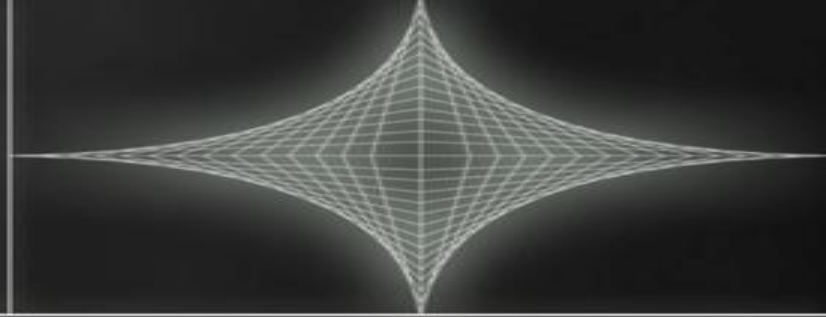
Statement of Purpose(Cont.)



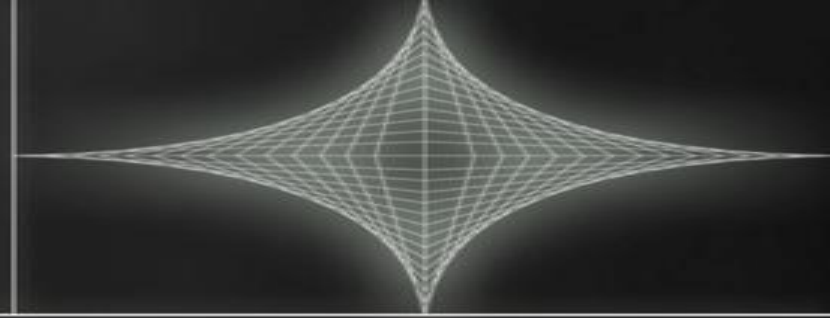
- ▶ 제약조건
 - ▶ 코드의 크기는 100~200줄 내외의 프로그램을 대상으로 한다.
 - ▶ Main Function을 포함하는 코드 여야 한다.
 - ▶ 단일 파일로 되어 있는 코드에 대하여 작동한다.
 - ▶ 사용자가 정의한 헤더를 사용한 파일에 대해서는 작동하지 않는다.
 - ▶ 포인터를 사용하지 않는 코드를 대상으로 한다.
 - ▶ 실행은 Cygwin 환경에서 이루어지며 GCC 컴파일을 이용한다.



System Context file Diagram



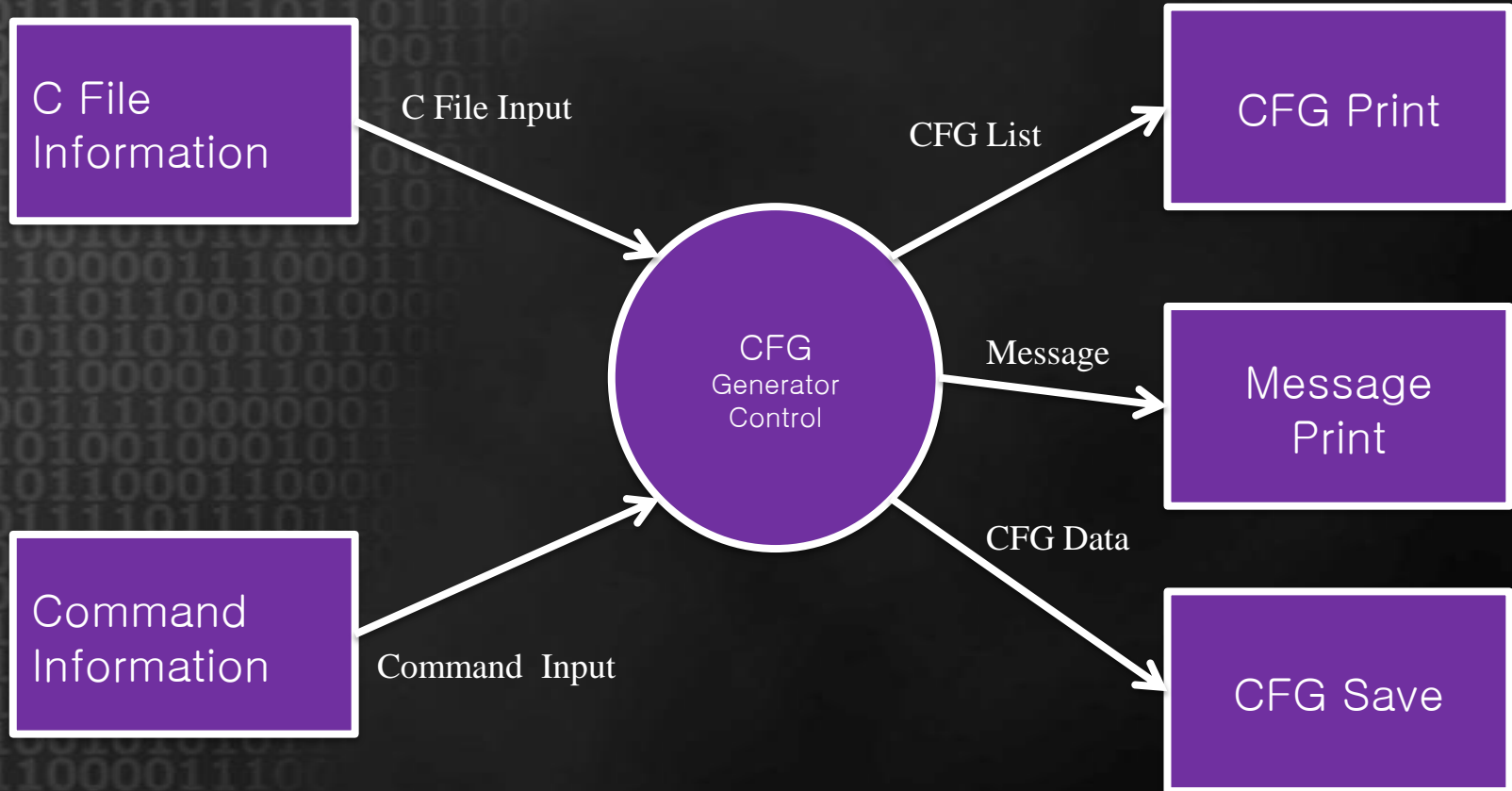
Event List



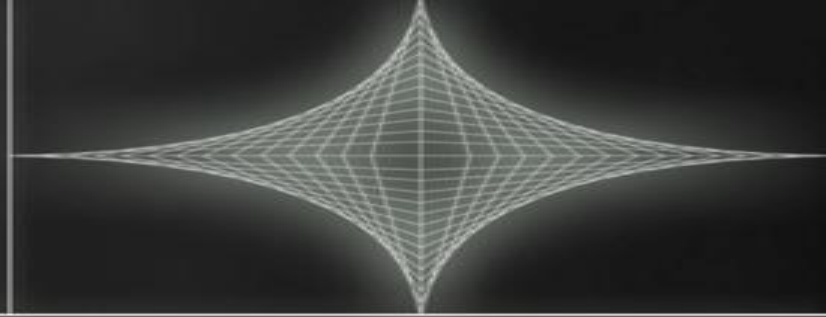
Input / Output Event	Description
C file Input	Input C file into CFG generator in order to make CFG
CFG List	Display CFG which is generated by CFG generator
Command Input	Input command that operates CFG generator
Message	Display essential system messages
CFG Data	Save CFG data from CFG generator in TXT format.



DFD Level 0



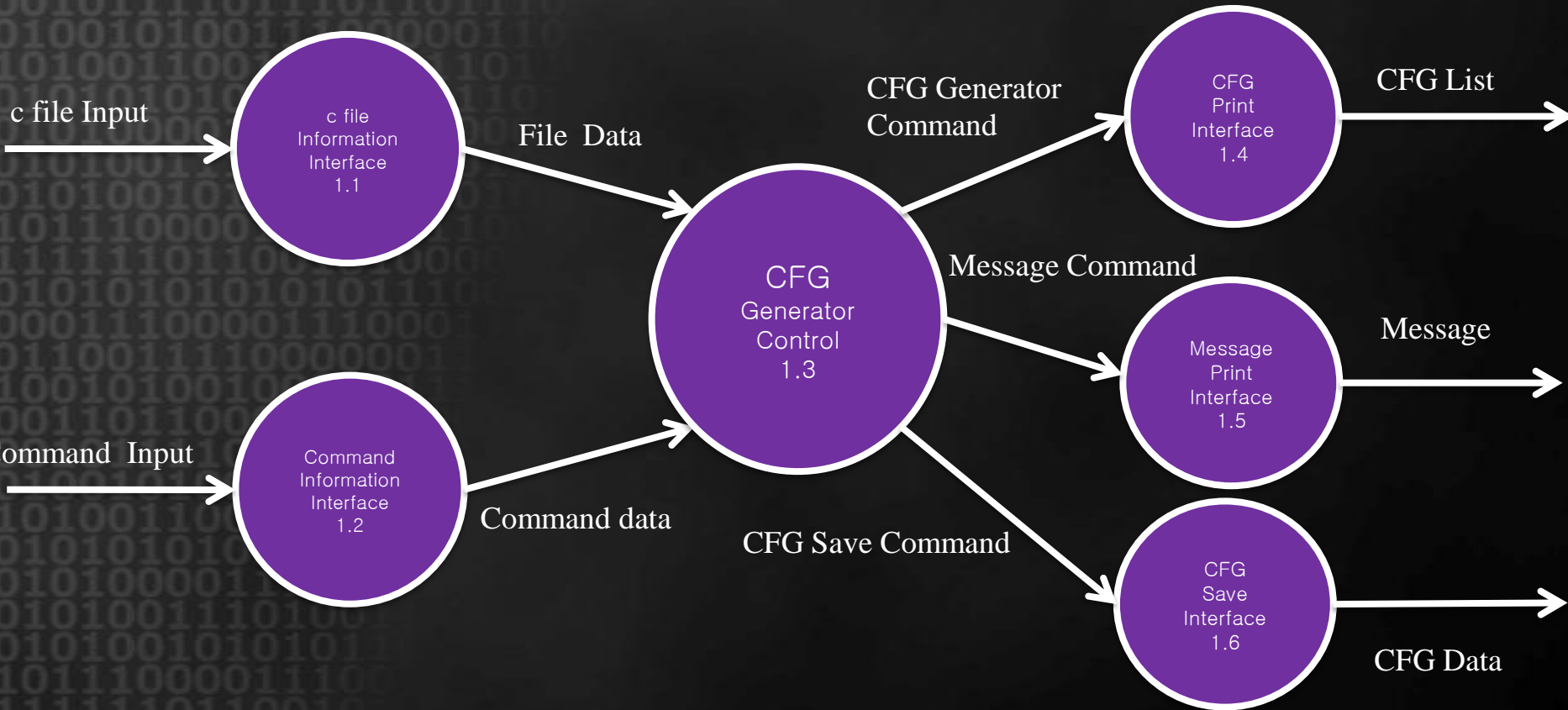
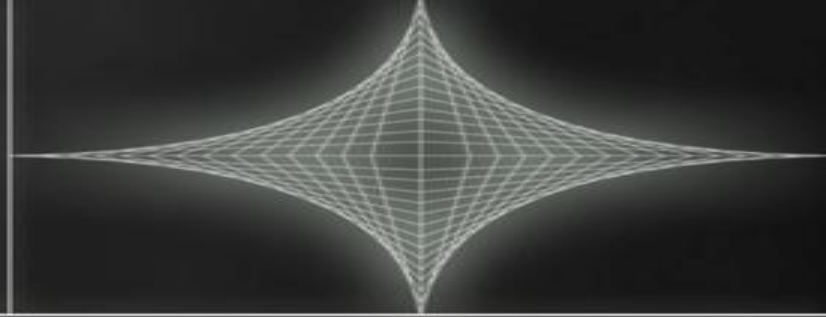
Data Dictionary



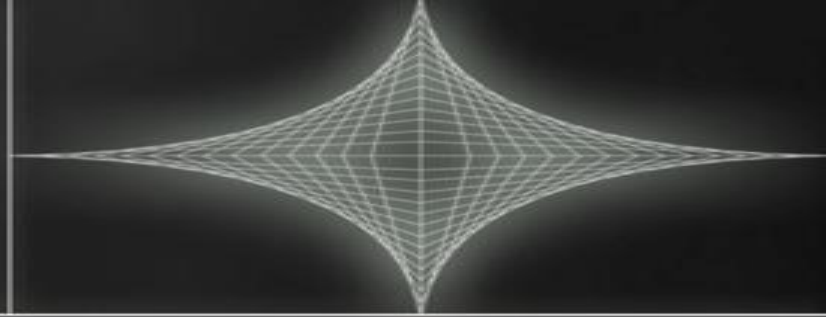
Input / Output	Description	Type
C file Input	Input C file into CFG generator in order to make CFG	C source file
CFG List	Display CFG which is generated by CFG generator	Double Linked List
Command Input	Input command that operates CFG generator	Char *
Message	Display essential system messages	Char *[]
CFG Data	Save CFG data from CFG generator in TXT format.	Txt file



DFD Level 1



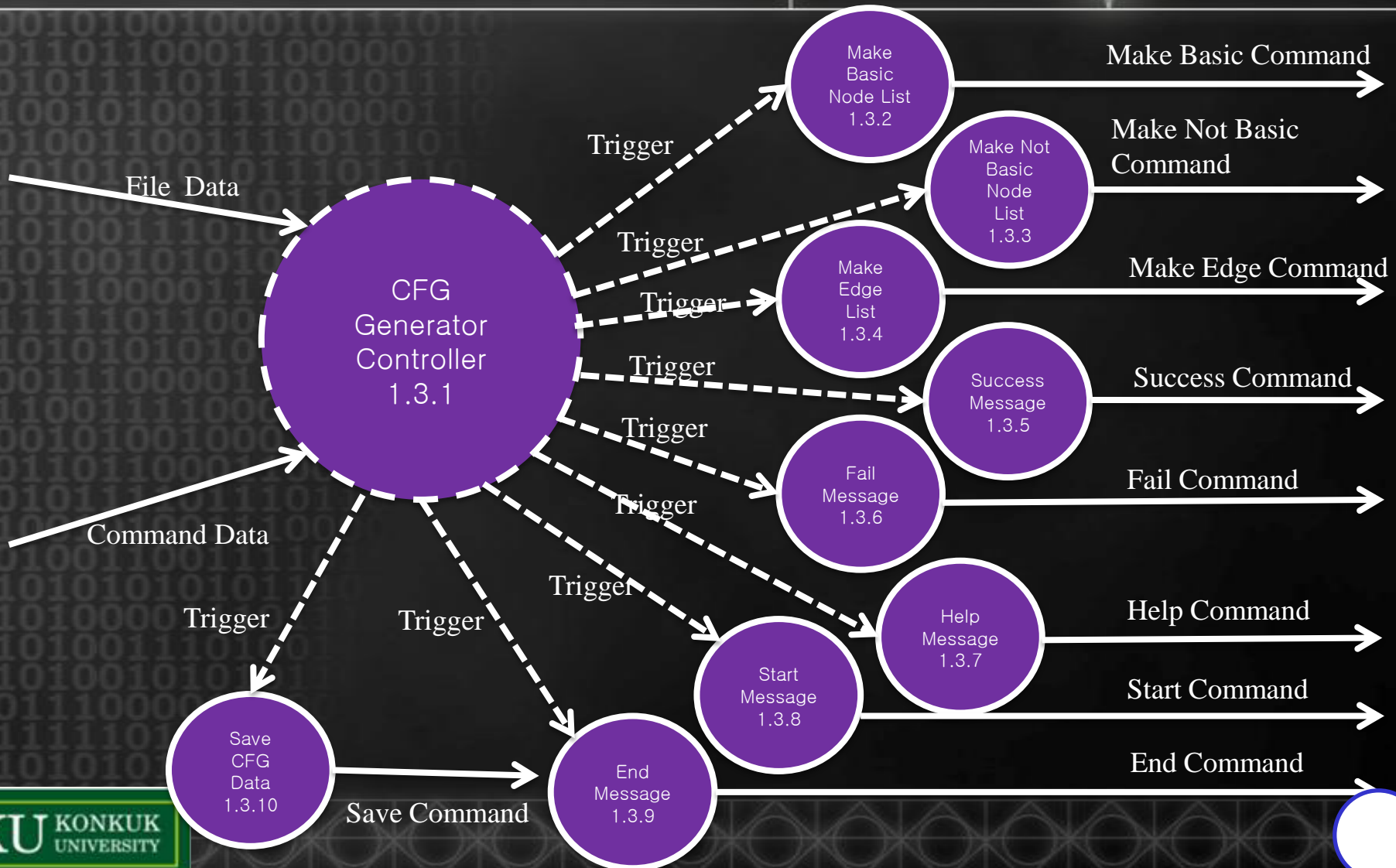
Data Dictionary



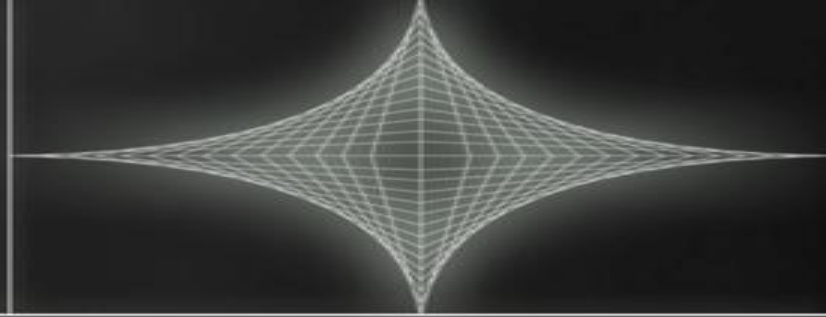
Input / Output	Description	Type
File Data	File pointer about content of the file	FILE*
Command Data	two validated data that are going to be used for names of the input and output data files	Char *[]
CFG Generator Command	Three sets of data structure for the creation of CFG list	Data Structure
Message Command	A set of messages that indicate current system status	Char *[]
CFG Save Command	Data structure which contains CFG data to be stored in TXT format	Data Structure



DFD Level 2

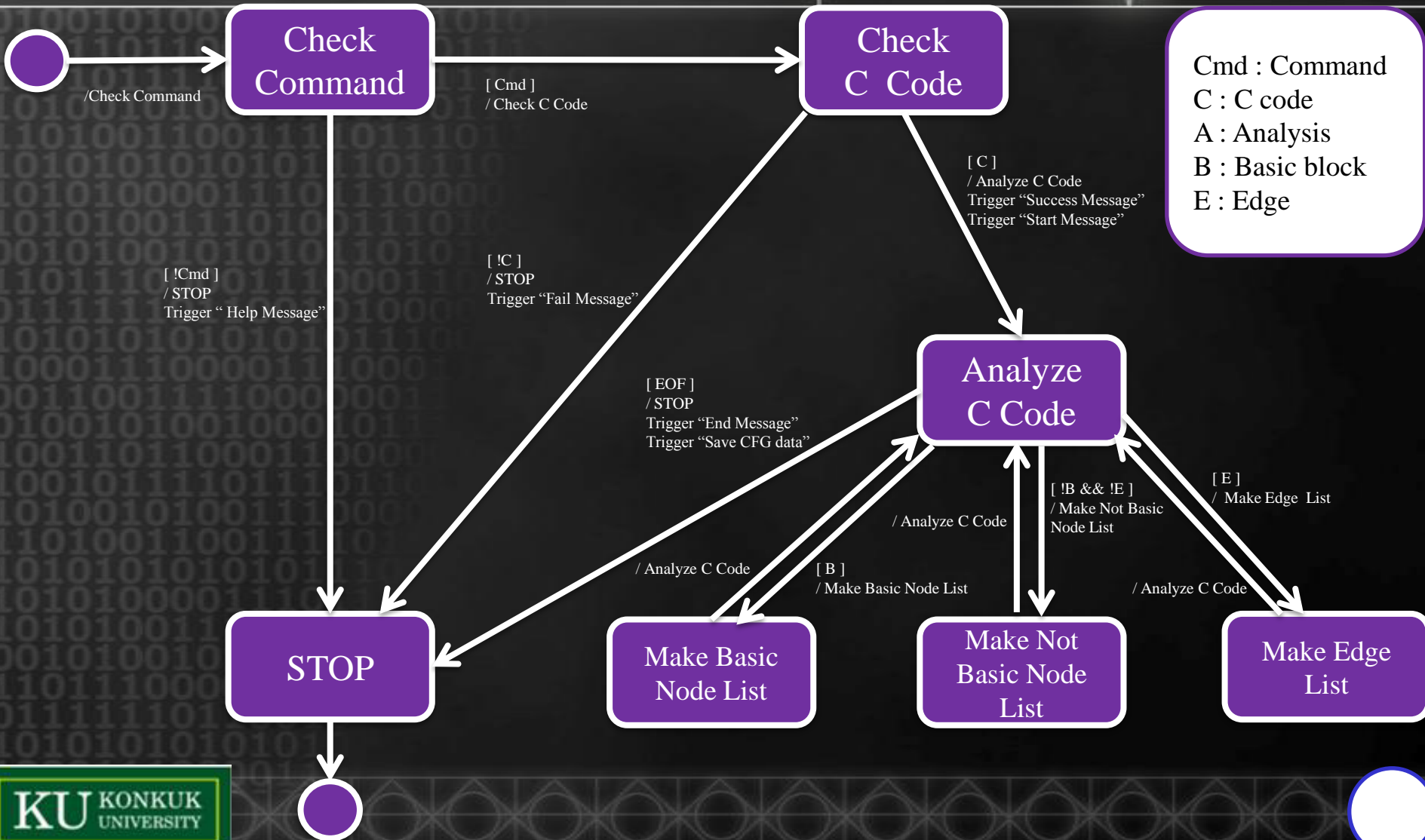


Data Dictionary

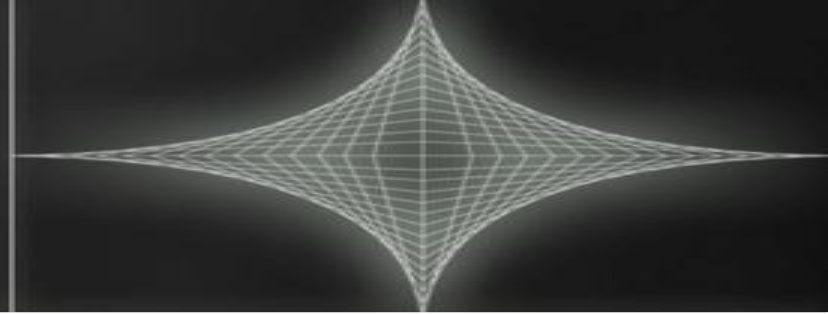


Input / Output	Description	Type
Make Basic Node Command	Add an expression statement column to the list with a number of the node	Data Structure
Make Not Basic Node Command	Add an iteration statement or selection statement column to the list with a number of the node	Data Structure
Make Edge Command	Add a column contains two nodes that are next to each other and their relationship	Data Structure
Success Command	Contains “Success” message that is used as a parameter in Message Print Interface	Char*
Fail Command	Contains “Fail” message that is used as a parameter in Message Print Interface	Char*
Help Command	Contains “Help” message that is used as a parameter in Message Print Interface	Char*
Start Command	Contains “Start” message that is used as a parameter in Message Print Interface	Char*
End Command	Contains “End” message that is used as a parameter in Message Print Interface	Char*
Save Command	A set of data that contains all the information about node and edge lists.	Data Structure

DFD Level 3

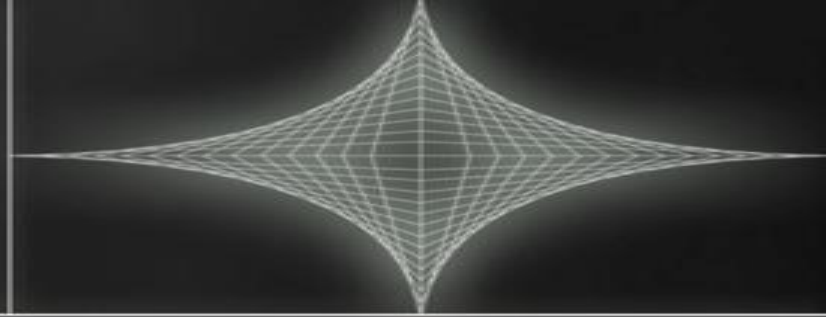


Process Specification



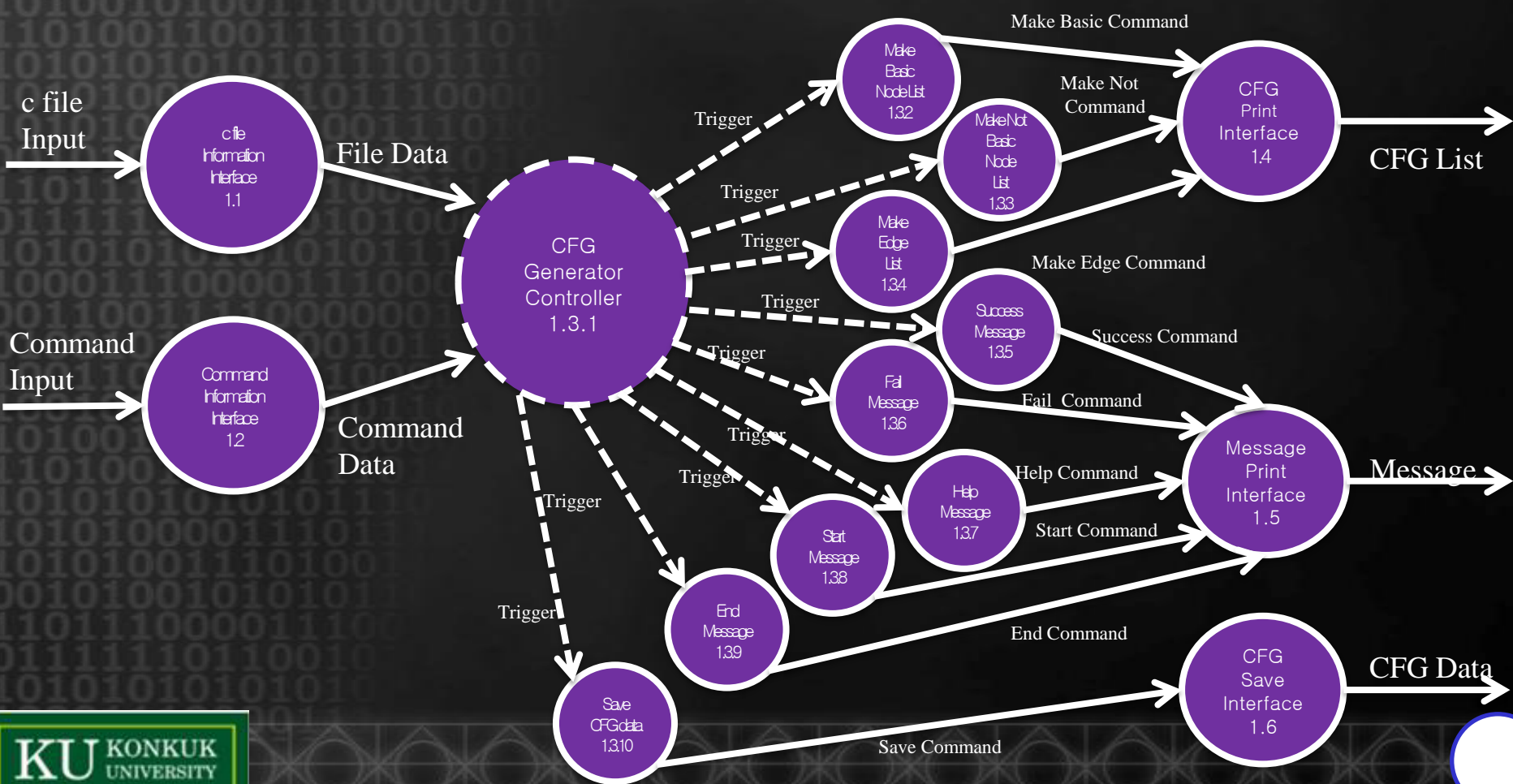
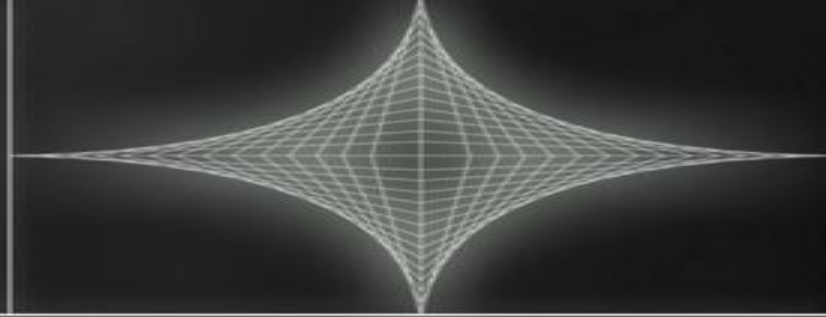
Process	Description
Check Command	<p>Check contents which were inserted from command line whether there are inappropriate keys or not. First argument column must be end with .c format and the second one with .txt format. If either of argument column has a problem, the system will be terminated with a “Help” message.</p>
Check C Command	<p>Check contents of the c file in order to get existence of the file, Line of Code, user-defined header file, presence of its main function, usage of pointers. If the system cannot find the file, or the file has less than 100 lines or more than 200 lines, or the file has user-defined header files or doesn't have main function, or the file uses pointers in its code, the system will be terminated with a “Fail” message. In contrast, if there's nothing wrong with the file, the system will show “Success” message and let users know the beginning of analyzation.</p>
Analyze C Code	<p>Examine each line of the code to determine its statement syntax. If the statement is expression statement, this process will call “Make Basic Node List” function in order to add a node into a basic-node list. If the statement analyzed as an iterating or selecting statement, the process will call “Make Not Basic Node List” function to add a node into a not-basic-node list. When the moment to make edge for nodes has come, the process will call “Make Edge List” function to insert relationship of the node into a edge list. If it's the End of File, the process displays “show” message and calls “Save CFG data”.</p>

Process Specification

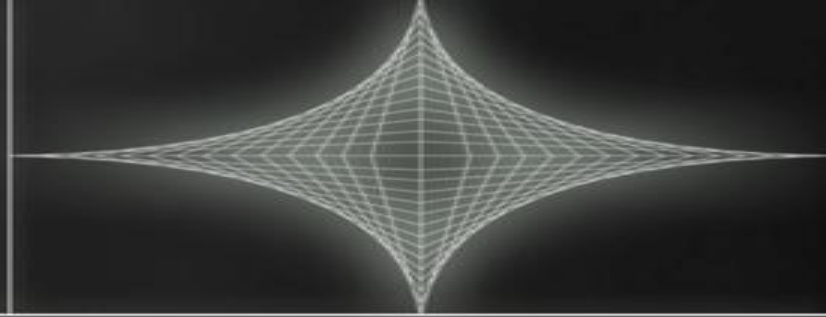


Process	Description
Make Basic Node List	Add a column to a basic-node list. Each column consists of contents of the expression statement and a number of node that increases by 1. the number of node starts from 0. After the action is over, the process calls “CFG Print Interface” to display statements on the screen.
Make Not Basic Node List	Add a column to a not-basic-node list. Each column consists of contents of the iteration or selection statement and a number of node that increases by 1. the number of node starts from 0. After the action is over, the process calls “CFG Print Interface” to display statements on the screen.
Make Edge List	Determine the relation of two nodes that are next to each other in the code and add a column to a edge list. The column contains locations of those nodes and their relationship. After the action is over, the process calls “CFG Print Interface” to display edges on the screen in certain forms following a specific rule.
Stop	Abstract Process that only shows the end of the main controller

DFD – Overall



Process Specification

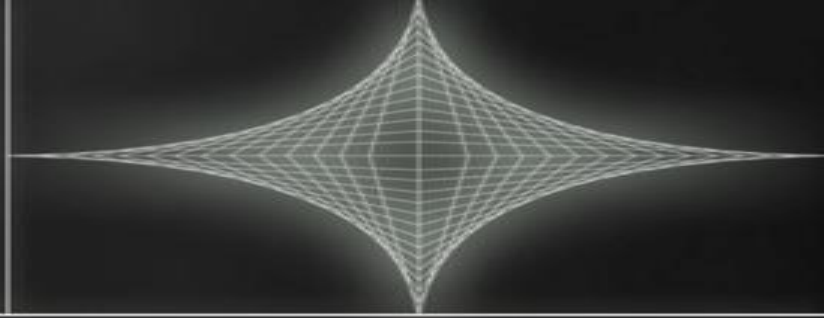


Reference No.	1.1
Name	c file Information Interface
Input	c file Input
Output	File Data
Process Description	After receiving c file Input, it sends File Data to CFG generator controller

Reference No.	1.2
Name	Command Information Interface
Input	Command Input
Output	Command Data
Process Description	After receiving Command Input, it sends Command Data to CFG generator controller



Process Specification

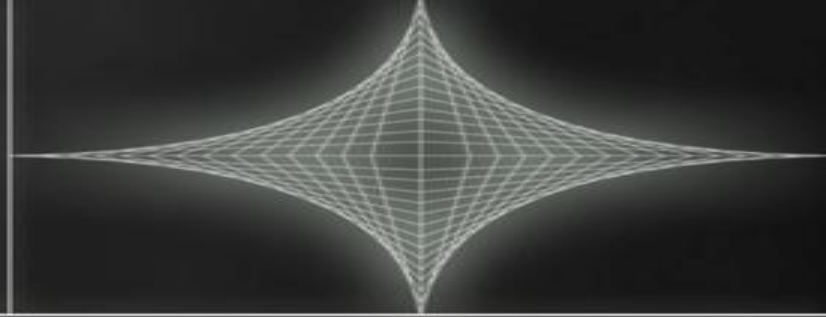


Reference No.	1.3.1
Name	CFG Generator Controller
Input	File Data, Command Data
Output	Trigger
Process Description	It is a main control that examines input data, procedures processes, and calls message functions.

Reference No.	1.3.2
Name	Make Basic Node List
Input	Trigger
Output	CFG Command
Process Description	Create a list of Basic Node.



Process Specification

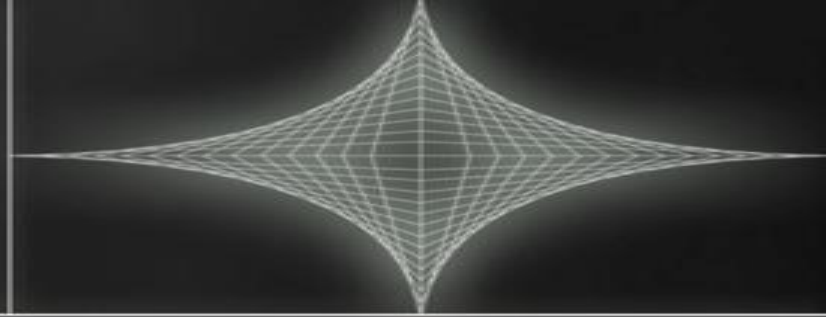


Reference No.	1.3.3
Name	Make Not Basic Node List
Input	Trigger
Output	CFG Command
Process Description	Create a list of Not Basic Node.

Reference No.	1.3.4
Name	Make Edge List
Input	Trigger
Output	CFG Command
Process Description	Create a list of edges.



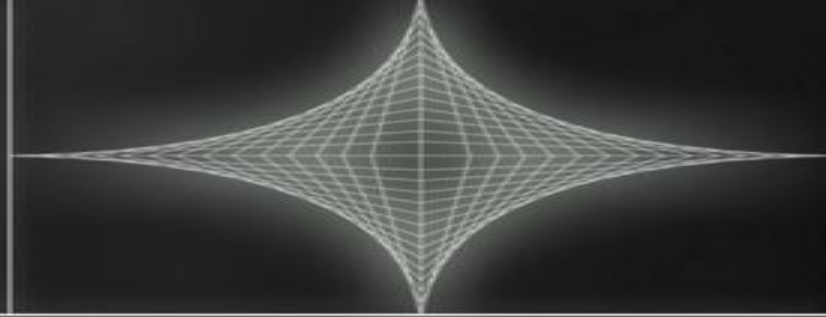
Process Specification



Reference No.	1.3.5
Name	Success Message
Input	Trigger
Output	Message Command
Process Description	A message is shown that indicates contents of the code are valid to be transformed into nodes and edges.

Reference No.	1.3.6
Name	Fail Message
Input	Trigger
Output	Message Command
Process Description	A message is shown that indicates contents of the code are not valid to be transformed into nodes and edges.

Process Specification

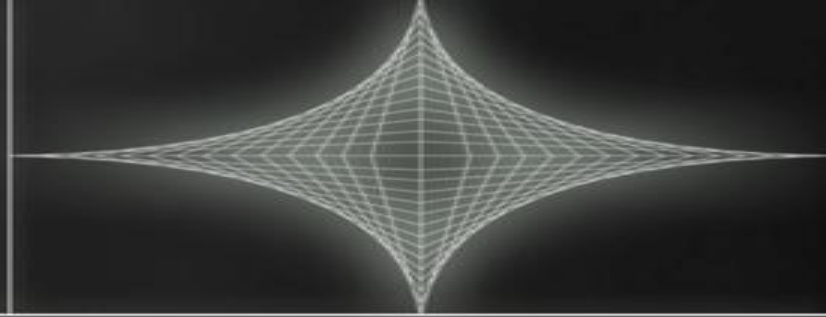


Reference No.	1.3.7
Name	Help Message
Input	Trigger
Output	Message Command
Process Description	A message is shown that indicates user entered an incorrect command.

Reference No.	1.3.8
Name	Start Message
Input	Trigger
Output	Message Command
Process Description	A message is shown that indicates the analyzation is going to begin.



Process Specification

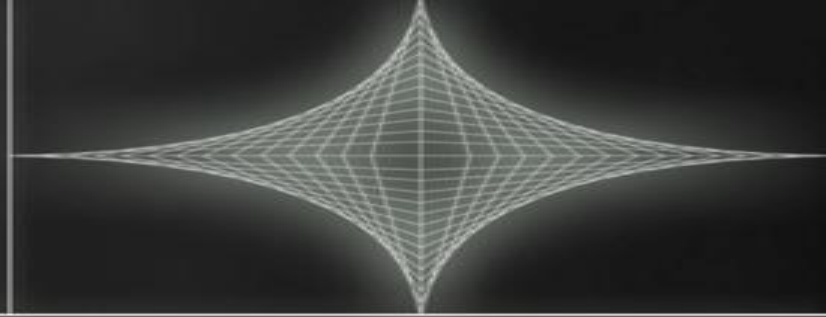


Reference No.	1.3.9
Name	End Message
Input	Trigger
Output	Message Command
Process Description	A message is shown that indicates the analyzation has been finished.

Reference No.	1.3.10
Name	Save CFG Data
Input	Trigger
Output	Save Command
Process Description	Integrates node and edge lists into one data structure so that it could be filed in TXT format.



Process Specification

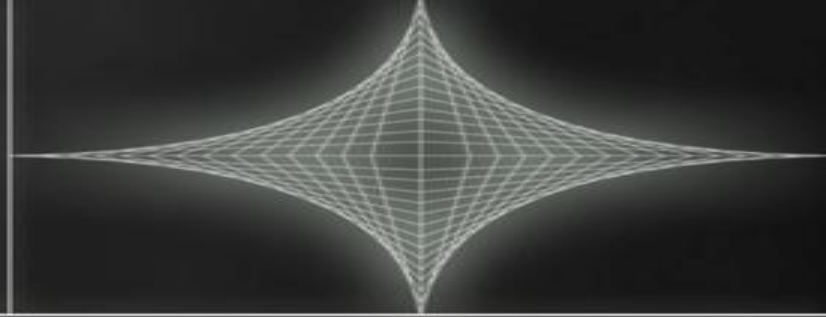


Reference No.	1.4
Name	CFG Print Interface
Input	CFG Command
Output	CFG List
Process Description	After receiving CFG Command, it sends CFG List to CFG Print in order to print CFG List

Reference No.	1.5
Name	Message Print Interface
Input	Message Command
Output	Message
Process Description	Display proper messages



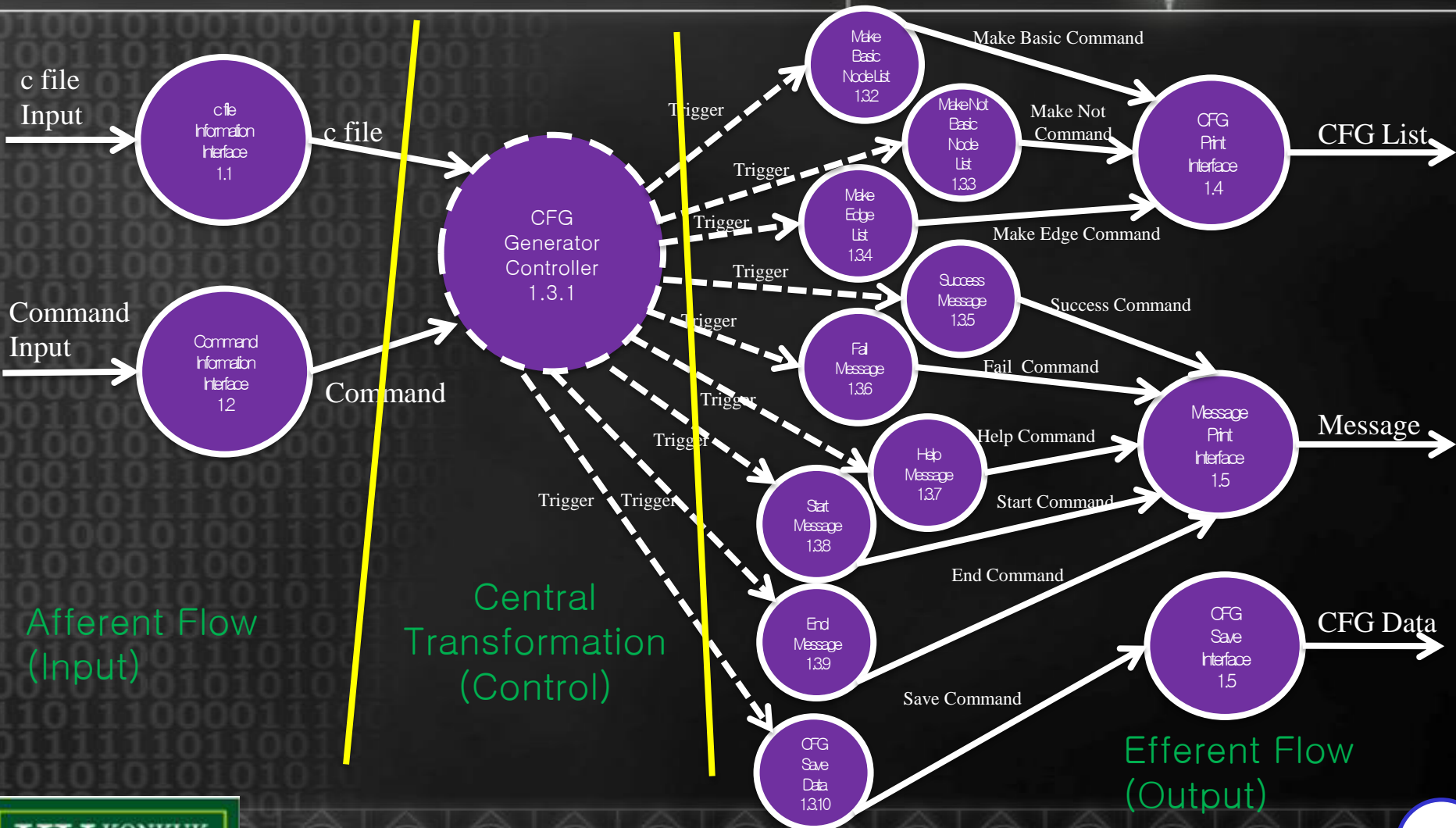
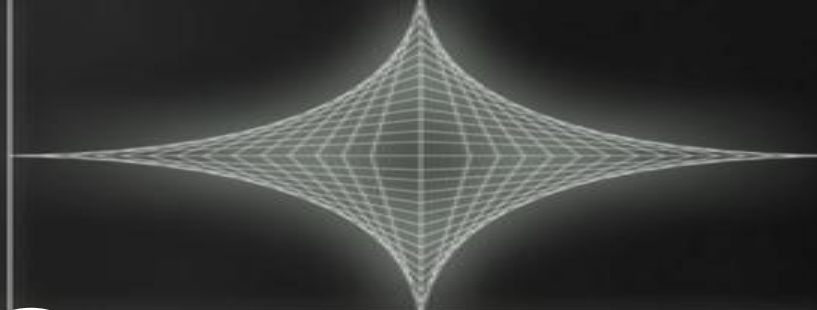
Process Specification



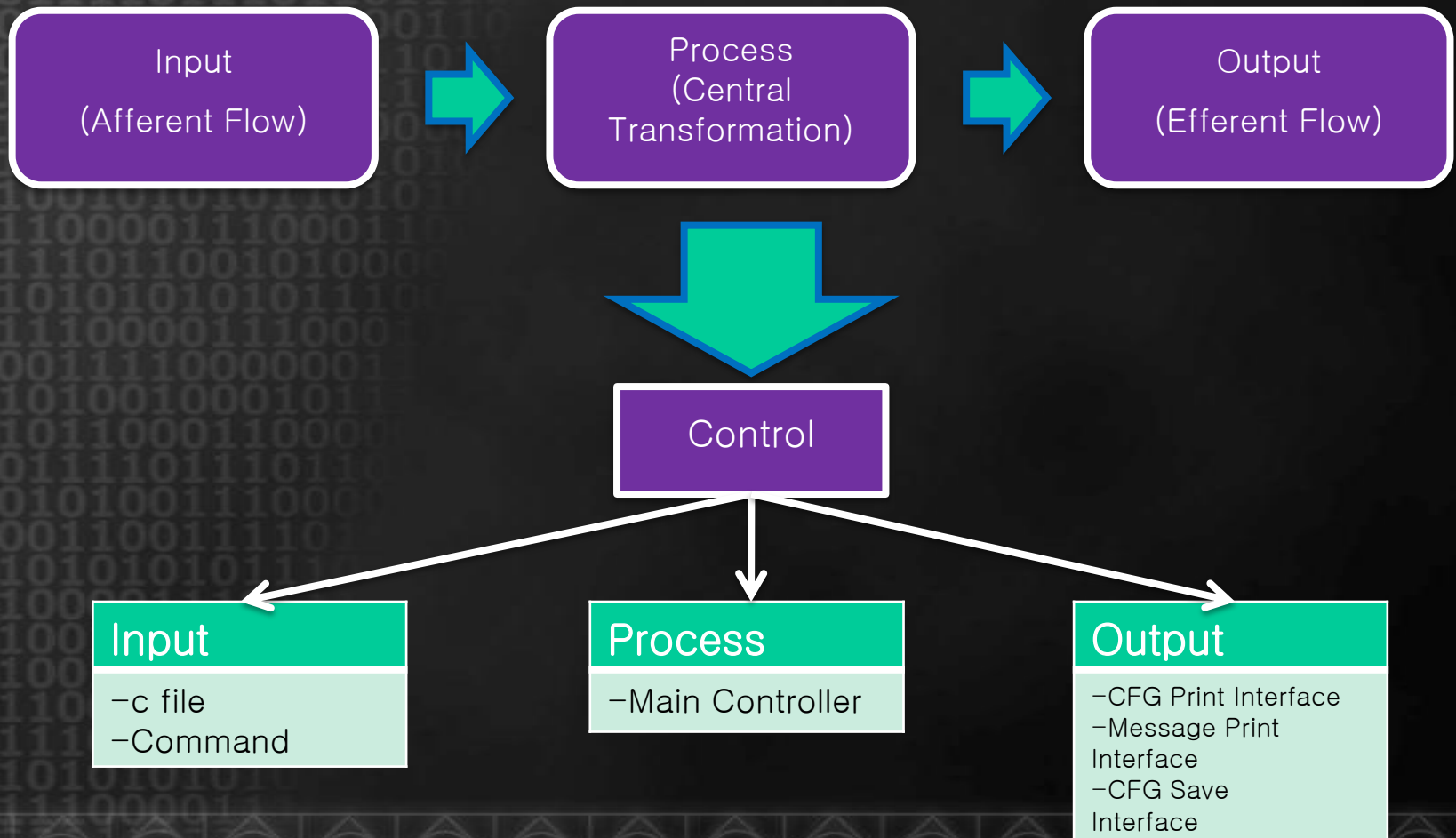
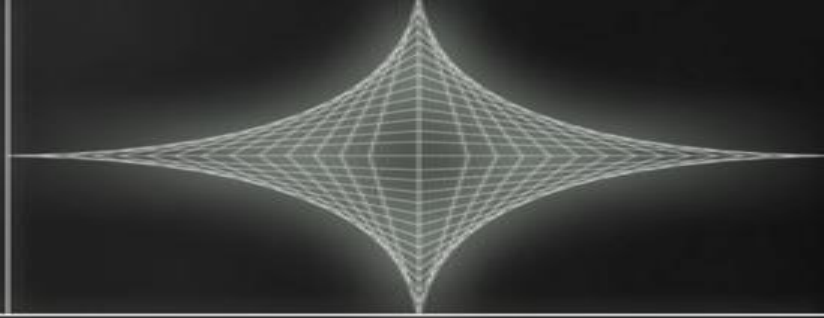
Reference No.	1.6
Name	CFG Save Interface
Input	Save Command
Output	CFG Data
Process Description	Transforms Save Command into CFG Data and saves it.



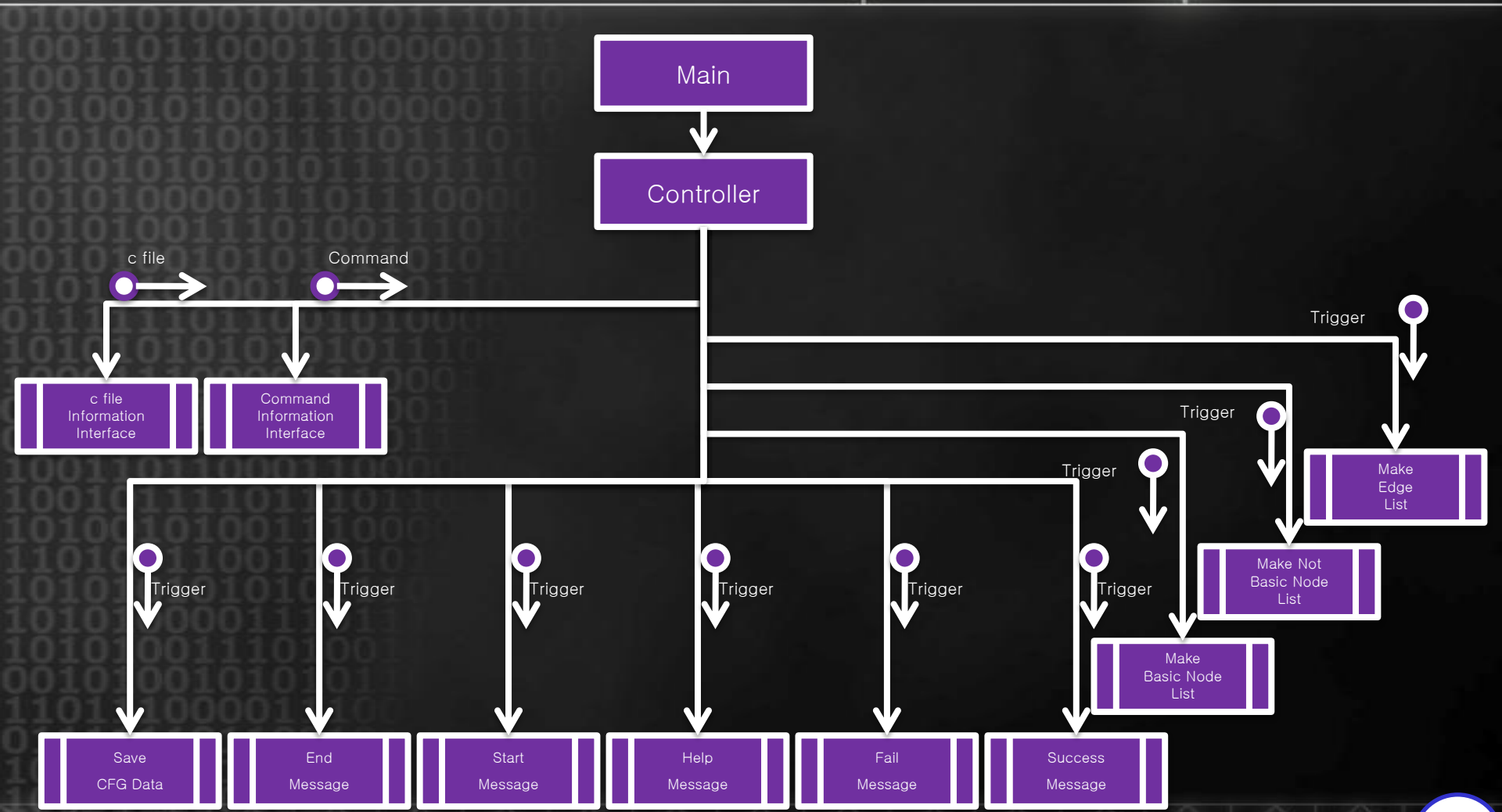
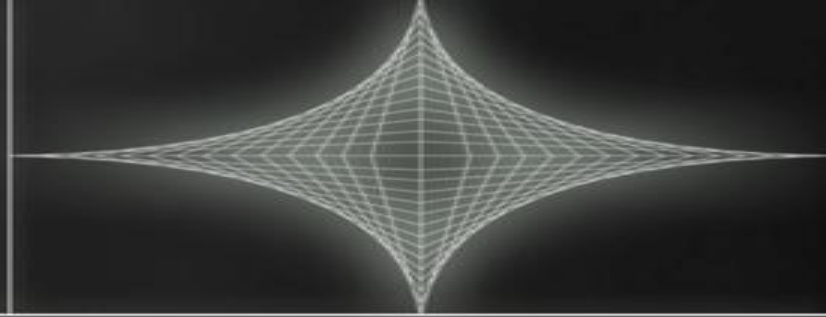
Structured Chart-Transform Analysis



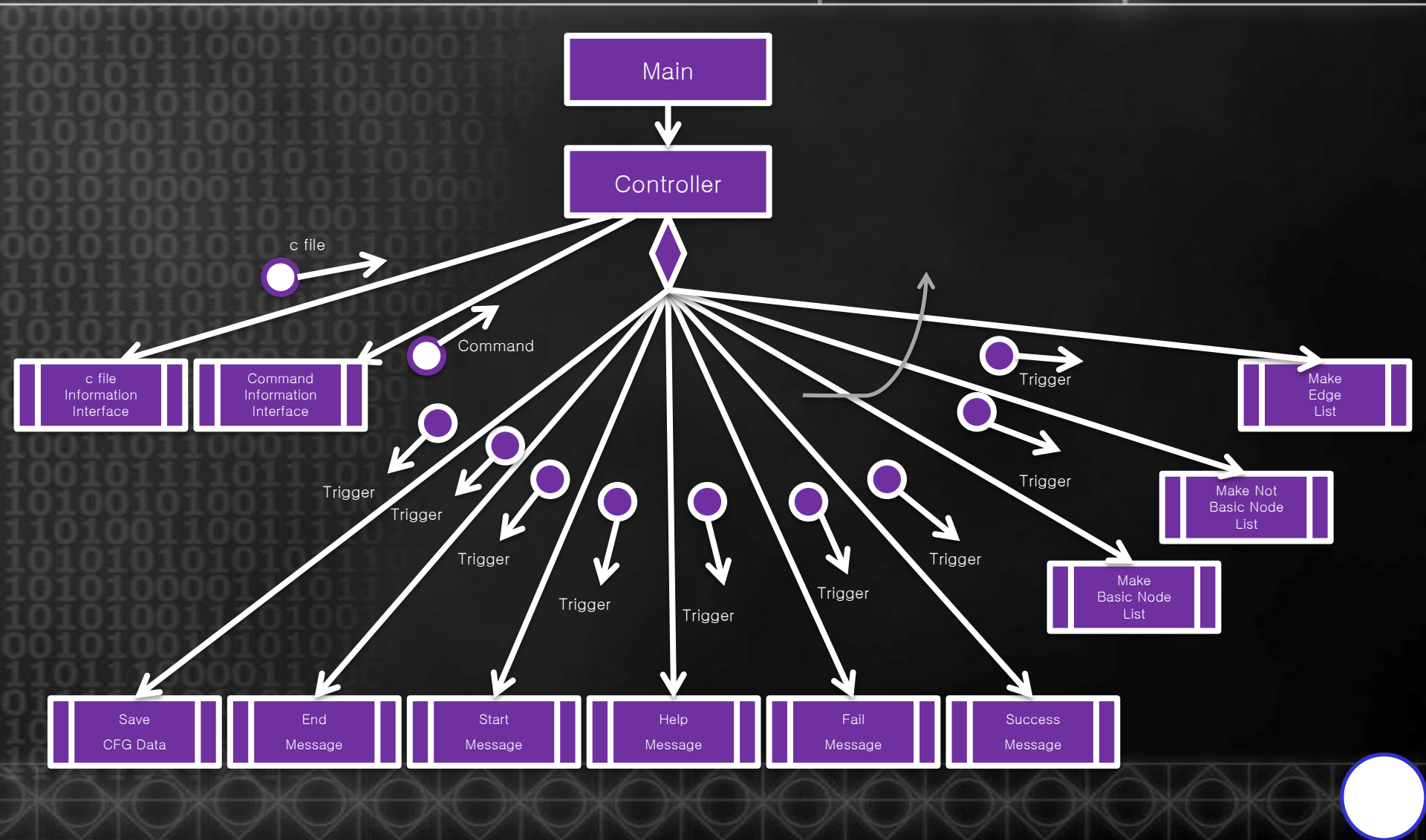
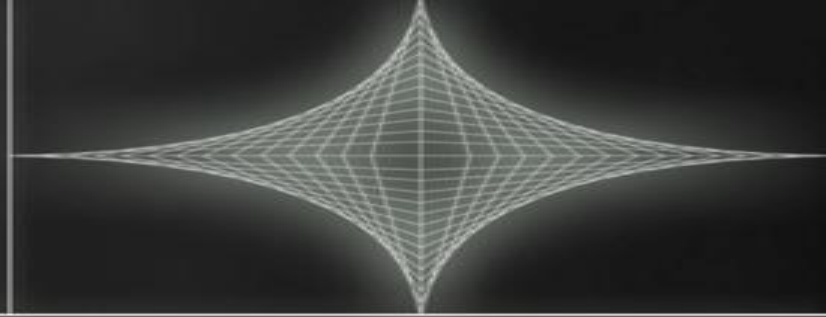
Structured Chart- Transform Analysis(Cont.)



Structured Charts – Basic



Structured Charts – Advanced



The End
Thank you