DESIGN AND DEVELOPMENT OF MOULD CALCULATION SOFTWARE

MUHAMAD IDZWAN BIN LATIF

Report submitted in partial fulfillment of
the requirements for the award of
Bachelor of Manufacturing Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: DESIGN AND DEVELOPMENT OF MOULDCALCULATION SOFTWARE

SESI PENGAJIAN: 2012/2013

Saya, **MUHAMAD IDZWAN BIN LATIF (900415-10-5693)**
(HURUF BESAR)

mengaku membenarkan tesis Projek Tahun Akhir ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi
   pengajian tinggi.
4. **Sila tandakan (√)

	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

_____________________ _____________________

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap:
Batu23, Jln Redan Hujung,
Kampung Kanchong Darat,
42700 Banting,Selangor.
Tarikh:

AHMAD ROSLI BIN ABDUL MANAF
(NamaPenyelia)
Tarikh:

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project report and in ouropinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Manufacturing Engineering.

Signature :

Name of Supervisor :EN. AHMAD ROSLI BIN ABDUL MANAF

Position : LECTURER

Date : 19JUNE 2013

STUDENT'S DECLARATION

I hereby declare that the work in this report is my own except for quotations and summaries which have been duly acknowledged. The report has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :

Name :MUHAMAD IDZWAN BIN LATIF

ID Number :FA 09031

Date : 19JUNE 2013

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and gratefulness to AllahAlmighty for His guidance and blessing me with the strength to complete this Final Year Project 1. I also would like to give my special thanks to my supervisor, En. Ahmad Rosli Bin Abdul Manaf for his guidance, advices and support all the way through the executionof this Final Year Project. Without their continued support, I would not have completed my Final Year Project. I also sincerely thank for the time spent proofreading and correcting my many mistakes.I is also grateful for my presentation panels, Dr. Noraini Binti MohdRazaliwho offer valuable recommendations and guides during the presentation of my project.

I acknowledge my sincere indebtedness and gratitude to my both beloved parents, Latif bin Sarman and Hasliah bt Katim for their love, dream and sacrifice throughout my life, and consistently encouraged me to carry on my higher studies in Malaysia. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Last but not least, I also would like to give my special thanks and appreciation to all my friends that helped and supported me a lot in this Final Year Project either directly or indirectly.

ABSRACT

Manual calculation for injection moulding operation is time consuming and intends to make a miscalculation because of many parameters that need to be considered. In this project, mould calculation software is developed and design. It is created to help the users to calculate the injection moulding parameters that involve in some processes. The software is developed and design by using Microsoft Visual Basic 2008 programming. The formulae of injection moulding parameters are input into the software, in the form of computer coding. The interfaces of injection moulding parameters are created by using the graphics in Microsoft Visual Basic 2008. At the end of the project, the Mould Calculations Software is created. The aid from software, time to calculate mould parameters in designing stage is shorter than manual calculation

ABSTRAK

Pengiraan secara manual untuk operasi acuan ini memakan masa dan berkemungkinan untuk membuat perhitungan yang salahkerana banyak parameter yang perlu diambilkira. Dalam projek ini, perisian pengiraan acuan dibangunkan dan direka bentuk. Ia dicipta untuk membantu pengguna untuk mengira parameter acuan yang terlibat dalam beberapa proses. Perisian ini dibangunkan dan reka bentuk dengan menggunakan Microsoft Visual Basic 2008pengaturcaraan. Formula acuan parameter input ke dalam perisian, dalam bentuk kod komputer. Muka parameter acuan dicipta dengan menggunakan grafik dalam Microsoft Visual Basic 2008. Pada akhir projek, PerisianPengiraan ParameterAcuan dicipta. Dengan bantuan daripada perisian, masa untuk mengira parameter acuan dalam peringkat mereka bentuk adalah lebih pendek daripada pengiraan manual

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Background

The project is about developing the plastic injection mould calculation software. Its involves parameters such as shot capacity, clamping pressure, plastic size capacity, calculation cycle time, number of cavities that determine by shot capacity, by plastic size capacity and by clamping capacity, gate and runner size . These criteria are important in injection moulding process.Sofware that used to develope the mould calculation sofware are visual basic and C++ programming. The result from the moulding calculation sofware will make the injection database more accurate. These two combination of sofware will make the mould calculation became more effective and save the time for doing calculation by the manually. At the end of the project, the sofware will be give a lot of benefit to the manufacturer especially in injection moulding process.

1.2 Problem Statement

Manufacturers may have some difficulties in finding commercial software for plastic injection mould parameter estimation which really suits their immediate use. Some existing software in the market has many restrictions in data entry. Even though the software is easy to use, users require a long time to become famaliar with the software. They are time consuming for learning, not user friendly if users do not well understand the meanings for individual input.

1.3 Objectives

The objectives that going to achieve in this system is:

i. To design mould parameter calculation sofware system.
ii. To develop mould parameter sofware system with the injection database.

1.4 Scope

The scope of the project is:

1 Developing the mould calculation software by using visual basic
2 Creating a database of the injection parameters.

1.5 Project Significance / Impact

There are few benefits that a user can gain from this software system such as it will savethe calculation time. Futhermore, the calculation will be more accurate. Besides that, this system also assist industrial practitioner in producing an accurate calculation for certain mould manufacturing projects. The usage of mobile devices results in improved data accuracy and increased mobility and convenience thereby streamlining movement and reducing human errors. Mobile devices increase productivity through reduced data entry efforts, reduced data capture activities, streamlined user time-motion efforts, enables process automation which increases throughput and decreases cycle-times.

CHAPTER 2

LITERATURE REVIEW

2.1 Mould

Mould is the common terms used to describe the tooling used to produce plastic parts in moulding. It's usually only used in mass production where thousands of parts were being produced. Typical moulds are constructed from hardened steel, pre-hardened steel, aluminum, and/or beryllium-copper alloy. The choice of material to build a mould from is primarily one of economics; in general, steel moulds cost more to construct, but their longer lifespan will offset the higher initial cost over a higher number of parts made before wearing out. Pre-hardened steel moulds are less wear-resistant and are used for lower volume requirements or larger components[1].

2.2 Mould Types

Mould is the production tooling used to produce plastic parts in moulding.There are two type of mould. There are **two-plate mould** and **three-plate mould.**Both of the mould type have different characteristic.

2.2.1 Two-Plate Mould

The two plate mould is simple in design, yet versatile. It consists of a front and stationary half. The cavity or core can be mounted on either half, depending upon the part design and the location of the location of the knock-out pins. This mould is easily adapted for different designs and all part ejection methods.The standard mould (Figure 2.1) is the most simple design, basically the standard moulds is same as two plate moulds construction, they divided in two side : cavity side and core side, cavity side is the side that construct to flowing plastic material from nozzle to cavity parts, basically they consist of sprue, runner.

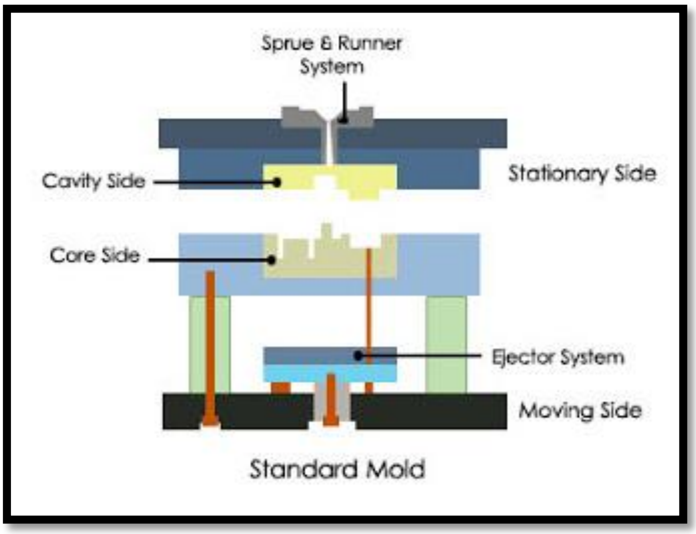


Figure 2.1: Two-plate mould

Core side constructed to make shape for core, demolding system and ejection system, at this side the design ejection system.Standard mould have one parting line, and have one opening direction. this type of mould use in all kinds of plastic parts that doesn't have undercut, inner and outer screw.Light brown color little and straight in ejection system is shown ejector pin.

2.2.2 Three-Plate Mould

This type of mould (Figure 2.2) is used primarily to centre gate or submarine gate parts in multiple cavities. It consists of the standard two-plate design with the third movable between the two. Automatic degating is possible with three-plate moulds, but runner scrap is increased.Basically three plate moulds has two parting line, and floating plate, floating plate support by support pin, Since the mould has two parting planes, the runner system can be located on one side of floating plate or make special plate that attach in floating plate, we called runner plate, see post about runner plate. Three plate moulds are used because of their flexibility in gating location. this types of moulds is flexible even use in multiple cavity.

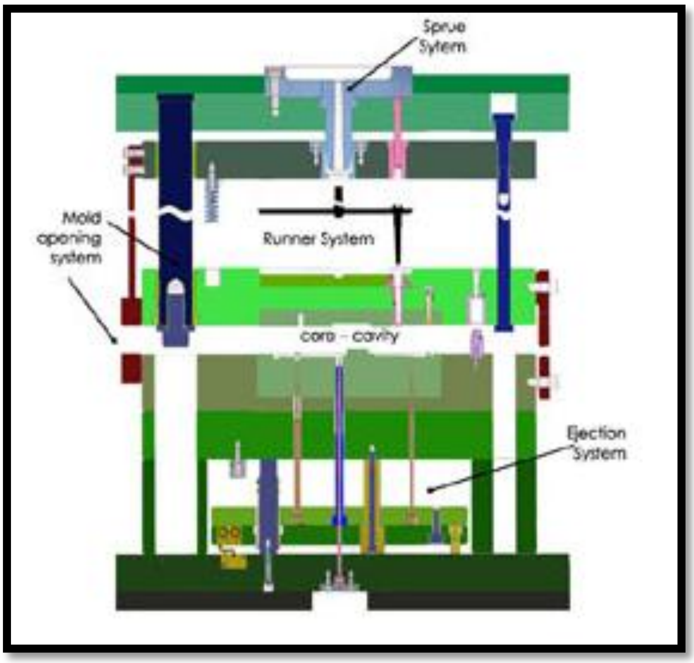


Figure 2.2:The Three plate moulds

2.3 Basic mould construction

TheFigure 2.3 below show Three-Plate Mouldbase type with closed position, basically 3 plate type and 2 plate type has some main plate.

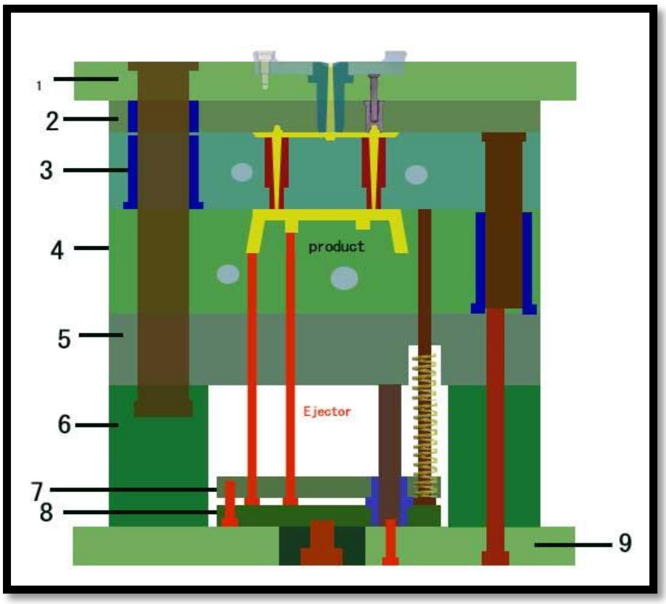


Figure 2.3:The Basic mould contruction

1. **Fixed Clamping Plate or Top Plate:** The function of top plate is to holds the fixed side of the mould to attached at the fixed platen of the injection machine at this plate will attach locating ring, eye bolt, and sprue bush.
2. **Runner Stripper Plate**: The runner stripper plateonly used in 3 plate moulds type, the function is to cut resin from nozzle in top of sprue bush, and pull the runner by runner locking pin.
3. **Fixed Mold Plate or Cavity plate:** The function of cavity plate is to hold cavity side of product, leader pin, support pin, Puller bolts, and Angular pin when slider attached.
4. **Movable Cavity Plate or Cavity plate**: It used to attach core side of product, return pin, leader bush and slider core if needed.

5. **Back up Plate or Support plate**: The function of support plate is to support cavity plate, attach the hole for return pin's spring, and cooling channel when in cavity plate can not make it.
6. **Spacer Block**: The spacer blocks are mounted between the movable clamping plate (bottom plate) and the movable cavity plate to give space and allow the ejector plate to move when ejecting thepart and the required length of spacer block depend on ejector stroke that needed to eject product.
7. **Ejector retainer plate**: The function of the ejector retainer plate is to hold the ejector, Z pin, shoulder bolts,and give space to ejector leader pin and support pillar.
8. **Ejector Plate**: The function of ejector plate is to pushes the ejector pins and return pins, connected with ejector rods.
9. **Movable Clamping Plate or Bottom plate**: The function of bottom plate is to holds the movable side of the mold like spacer block, support plate, cavity plate and ejector mechanism to the movable platen of the injection machine.

2.4.1 Sprue Bush

The sprue bush is defined as that part if the mould in which the sprue is formed. In practice (Figure 2.4) the sprue bush is the connecting member between the machine nozzle and the mould face, and provides suitable aperture through which the material can travel on its way to the impression or to the start of the runner system in multi-impression moulds.

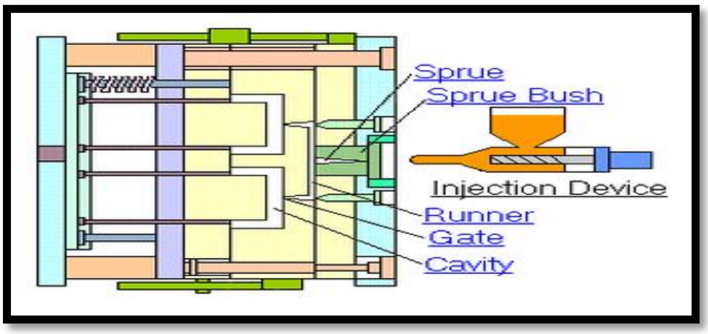


Figure 2.4: The location of sprue bush

The sprue bush is fairly highly stressed in some appliance and should therefore be made from a 1.5% nickel chrome steel and should always be hardened. The internal aperture of the sprue bush has between 2˚ and 4˚ included taper, which facifilitates removal of the sprue from the mould at the end of molding cycle.

There are two basic design of sprue bush. There are **spherical recess**and**flat rear face**as shown in Figure 2.5 :

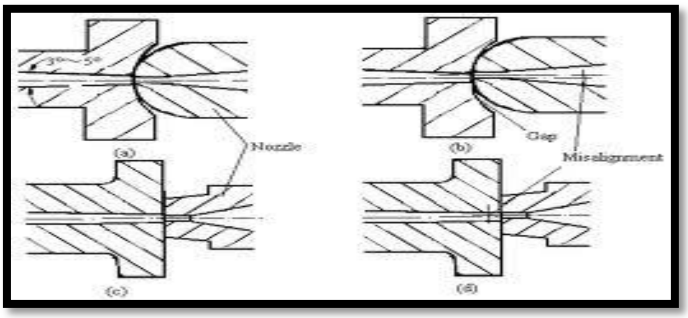


Figure 2.5: Two basic design of sprue bush,spherical recess and flat rear face

1. The first of these is a sprue bush with a **spherical recess** which is used in conjunction with a spherical front ended nozzle.
2. The second has a perfected **flat rear face**, the seating between it and it corresponding nozzle.

2.4.2 Register ring / Locating ring

The register ring is a circular member fitted on to the front face (and often also to the rear face) of the mould. Its purpose is to register (or locate) the mold is correct position on the injection machine.
When the mould is mounted n the machine, the front mounted register ring fits into a circular ole which is accurately machined in the injection platen on the cylinder-nozzle axis. This ensures that the small aperture in the nozzle is in direct alignment with the sprue bush. Now, since the sprue bush is the connecting member between

the machine nozzle and the mould face, this alignment of nozzle aperture and sprue bush hole permits an uninterrupted flow of material from the cylinder, through the nozzle and sprue hole into the mold runner system. The register ring, in fact, forms a direct connection between the sprue bush and the hole in the injection platen of the machine (Figure 2.6).

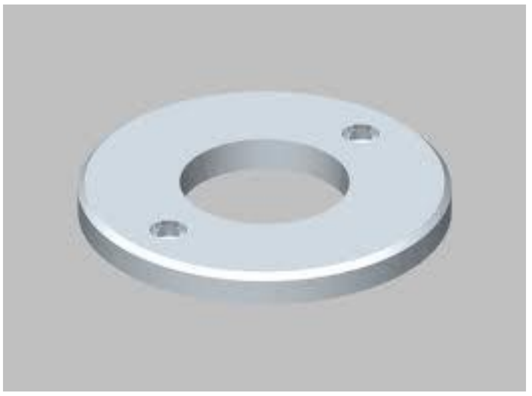


Figure 2.6: The Register ring / Locating ring

2.4.3 Runner

A Runner is a machined groove located between (and includes) the Sprue Bushing and the Gate. The function of a runner is to provide a passage for the material to flow from the Sprue Bushing to the Gate. There are many types of Runner cross sectional shapes. Most common shapes are the Full Round, Half Round, and the Trapezoidal. The length of a runner system should be kept to a minimum. Injection Pressure build-up due to long runner lengths can be reduced by increasing the runner diameter. However, larger runner diameters increase cycle time due to the added volume of material that needs to be chilled/solidified.

There are two types of runner:

1) Cold Runner

2) Hot Runner

2.4.3.1 Cold Runner

This runner system have similar use with hot runner, but in different temperature, the temperature between 80-120 C, this Cold runner system used in reactive material such us rubber and thermosets type of plastic.

The difficulties in Cold runner system is :

1. High pressure consumption, so design more expensive
2. Because material is reactive, a little different temperature make different viscosities, this will make every cavity have different time to full fill.

2.4.3.2 Hot Runner System

This system (Figure 2.7) should be viewed as extended nozzle in form of block, the hot runner system contains : sprue bush, runners, gate and extended nozzle, the advantage of hot runner is:

- They can completely eliminate runner scrap, so there are no runners to sort from the parts,
- No runners to throw away or regrind and remix into the original material.
- No loss of melt and thus less energy and work input
- Easier for fully automatic operation
- Very good quality because melt can be transfered into the cavity at the optimum site

The hot runners also have their own disadvantages and there are :

- High cost
- Thermal isolation from the hot runner manifold block is problematic

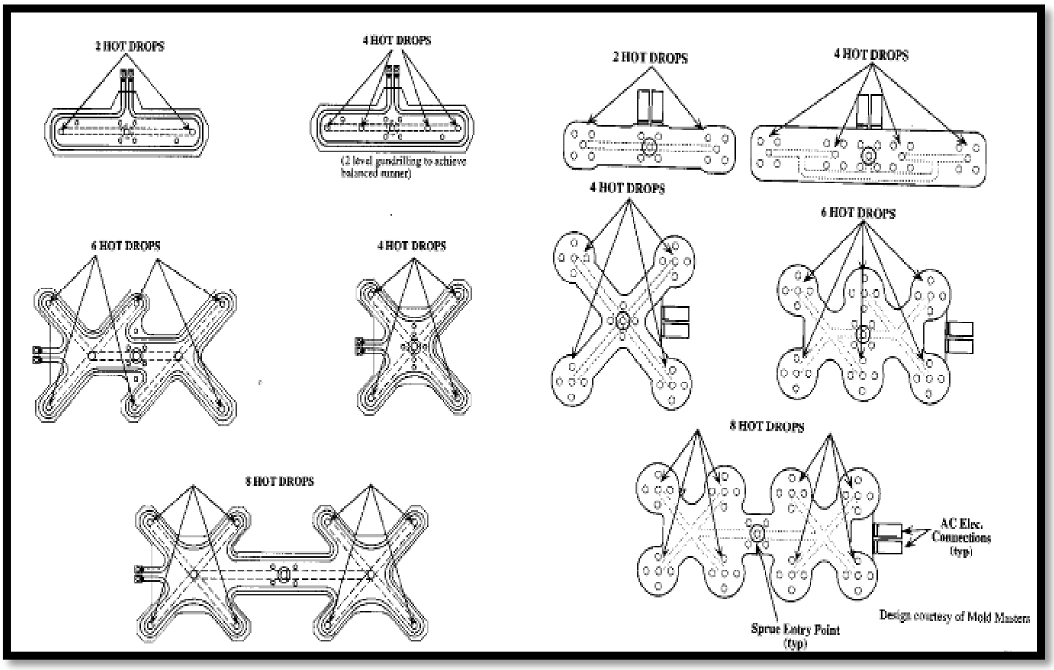


Figure 2.7: The hot runner system

The hot runners are popular in high production parts, especially with a lot of cavities, there is 3 type of Hot Runner are :

1. the insulated hot runner,
2. the internally heated hot-runner system,
3. the externally heated hot-runner system

2.4.4 Gates system

The narrow and shallow portion of the runner as it enters the cavity is called the gate.

Listed here are the different types of gate with their characteristics :

- Side gate

Side gate (Figure 2.8) is the most commonly used gate type and is commonly used for mould structures with 2 or more cavities. It is placed at the side of the plastic product. The gate has to be cut manually by a cutter.

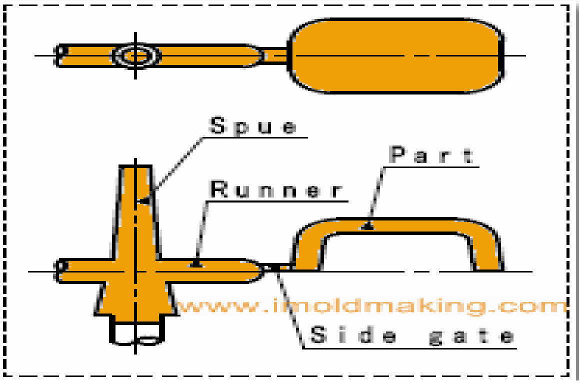


Figure 2.8 : Side gate

- Submarine gate

The positioning of this gate (Figure 2.9) is flexible thoughout the sides of the plastic product. It can be placed on the fixed or movable side of the mould but the design has to be thought about carefully so that the product will not be left inside the fixed cavity. The gate automatically cuts itself as the mould opens.

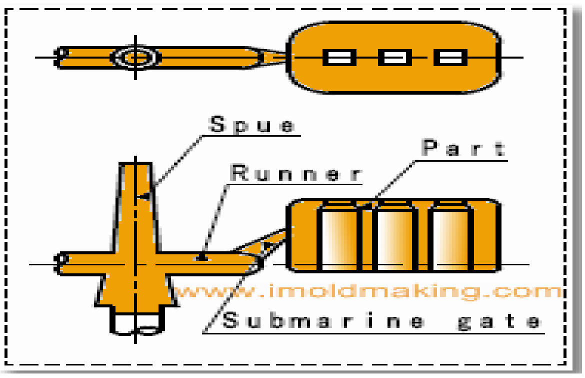


Figure 2.9 : Submarine gate

- Fan gate

Fan gate (Figure 2.10) used for large and flat plate products. It is placed at the side of the product – same as the side gate. The gate has to be cut manually by a cutter.

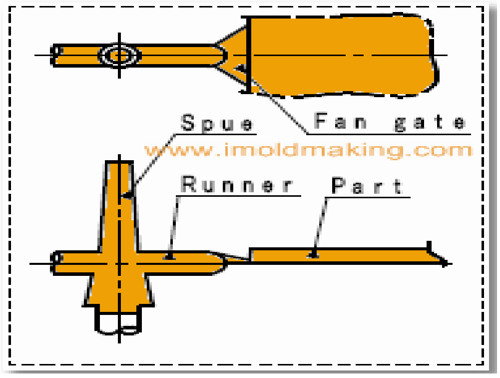


Figure 2.10 : Fan gate

- Pin gate

Pin gate is possible for molding multiple cavities or parts. The gate positioning (Figure 2.11) is relatively flexible at the top side of the product. The runner layout is very flexible as well. The mould base structure is complicated because it uses a 3-plate method.

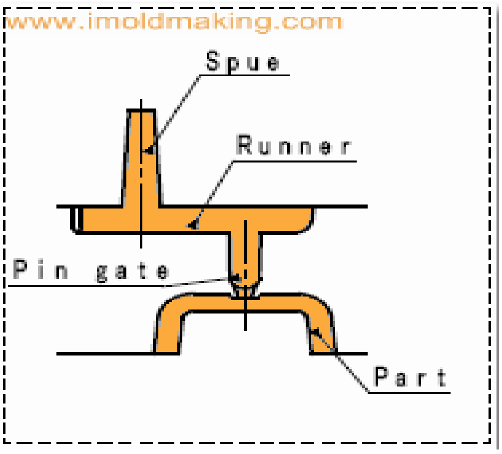


Figure 2.11: Pin gate

2.5 Injection Moulding Process

The mould consists of two primary components, the injection mold (A plate) and the ejector mould (B plate) as shown in Figure 2.12.

Figure 2.12 : The core and cavity plate

Plastic resin enters the mould through a spruein the injection mould; the sprue bushing is to seal tightly against the nozzle of the injection barrel of the moulding machine and to allow molten plastic to flow from the barrel into the mould, also known as the cavity. The sprue bushing directs the molten plastic to the cavity images through channels that are machined into the faces of the A and B plates. These channels allow plastic to run along them, so they are referred to as runners. The molten plastic flows through the runner and enters one or more specialized gates and into the cavity geometry to form the desired part. The amount of resin required to fill the sprue, runner and cavities of a mould is a shot [1].

2.6 Injection molding machine

An injection moulding machine, also known as an injection press, is a machine for manufacturing plastic products by the injection moulding process. It consists of 18 parts in injection moulding machine as shown in Figure 2.13.

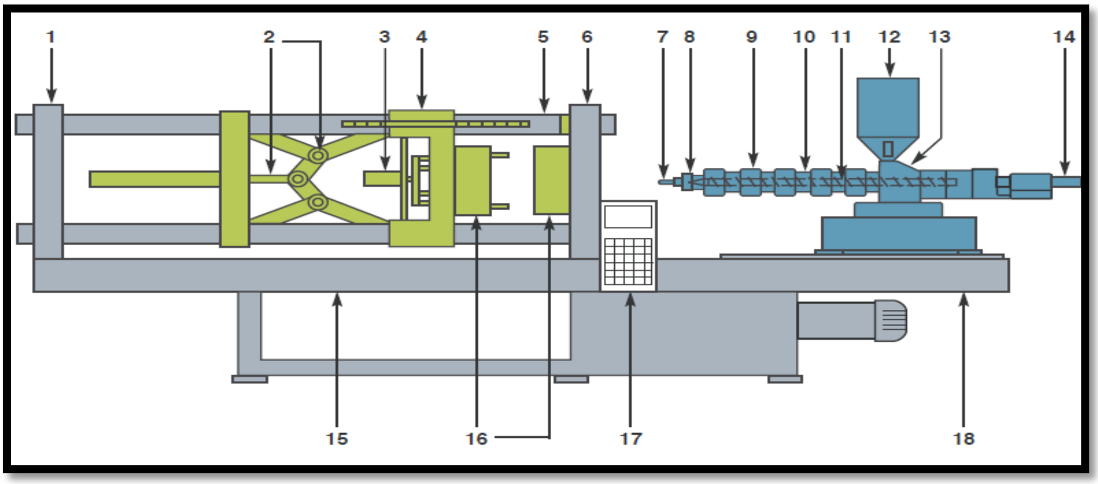


Figure 2.13: Injection molding machine

1. Backing platen
2. Closing mechanism - Toggle lever and cylinder
3. Ejector
4. Movable platen (Floating platen)
5. Tie bar
6. Fixed platen
7. Nozzle
8. Barrel head
9. Heater band
10. Injection barrel (Transfer chamber)
11. Screw
12. Feed hopper
13. Feed throat
14. Screw motor
15. Parts discharge opening
16. Mold
17. Digital control panel
18. Frame

Injection moulding is the most important moulding method for thermoplastics. It is based on the ability of thermoplastic materials to be softened by heat and to harden when cooled. The process thus consists essentially of softening the material in a heated cylinder and injecting it under pressure into the mould cavity, where it hardens by cooling. Each step is carried out in a separate zone of the same apparatus in the cyclic operation. As for the injection moulding machine, several types such as plunger type, plunger preplasticating type, screw preplasticating type and in-line screw type, etc. have been developed so far, but presently the in-line screw type injection moulding machine[5].

2.7 Machine operation

A diagram of a typical injection-molding machine is shown in Figure 2.11. Granular material (the plastic resin) falls from the hopper into the barrel when the plunger is withdrawn. The plunger then pushes the material into the heating zone, where it is heated and softened (plasticized or plasticated). Rapid heating takes place due to spreading of the polymer into a thin film around a screw. The already molten polymer displaced by this new material is pushed forward through the nozzle, which is in intimate contact with the mould. The molten polymer flows through the sprue opening in the die, down the runner, past the gate, and into the mould cavity. The mould is held tightly closed by the clamping action of the press platen.

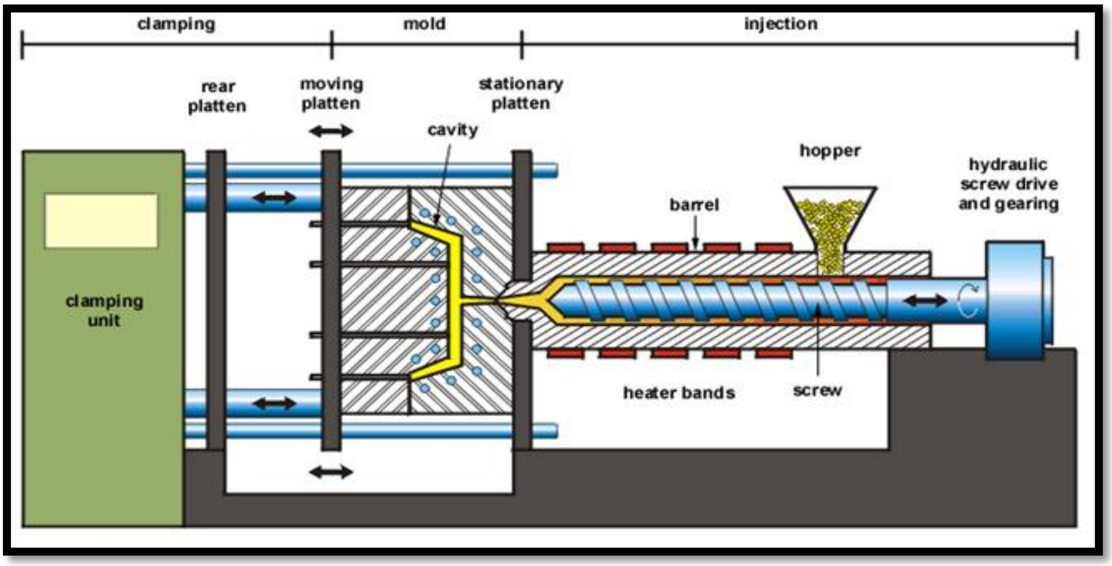


Figure 2.14: Injection molding machine

The molten polymer is thus forced into all parts of the mould cavities, giving a perfect reproduction of the mould. The material in the mould must be cooled under pressure below Tm or Tg before the mould is opened and the moulded part is ejected. The plunger is then withdrawn, a fresh charge of material drops down, the mould is closed under a locking force, and the entire cycle is repeated.

2.8 Calculation

This software system will calculate the parameter for plastic injection mould which is including calculation for **shot capacity** for plunger type and screw type**, plasticizing capacity**, **cycle time**, **clamping force**, **number of cavity, gate and runner size**.

2.8.1 Shot Capacity

Shot Capacity is a value of maximum weight of molten resin that the injection molding machine can push out with one forward stroke of the screw during injection molding operation. If the weight of molten resin is too high in the part, the part will has some kind of problem such as it will be over pack. And if the weight of molten is not enough or low than expected, it also will has some kind of problem, for example the part will become rejected part.

For plunger type: shot weight capacity with material B

= shot weight capacity with material A× (density B / density of A) ×
(bulk factor of A / bulk factor of B)

For screw type:

Shot capacity (g) = swept volume × (r) density × c

c = 0.85 for crystalline plastics

c = 0.93 for amorphous plastics acrylic

This rating depends both on the volume of the cylinder swept during one stroke of the plunger and on the volumetric capacity of the feed mechanism. Thus its determination requires correcting for the bulk density of the mix. A preferable way to express shot capacity is in term of the volume of material that can injected by the plunger into a mould at a specific pressure.

2.8.2 Plasticizing capacity

One factor affecting the output of a screw-type moulding machine is its plasticizing capacity. This is usually defined as the weight per hour (expressed in kg/hr) of material than can be heated to the molding temperature. Many plunger machines are fitted with a torpedo to improve the uniformity of the temperature in the mix and with preplasticizing units to increase the plasticizing capacity. The torpedo, which is in the heated cylinder, forces the viscous mix into closer contact with the cylinder walls. Its fins act to conduct heat from the hot cylinder walls to the torpedo. In some machines the torpedo has a separate heating unit. The plasticizing capacity can affects the volume of the molded article that can be prepared. Therefore, in the case of a molded item that requires a large volume of plastic, it is necessary to prepare a plasticizing unit having a large plasticizing capacity.

Plasticizing rate of material B kg/hr:

= plastics rate of material A (kg/hr) × (specific heat capacity A / specific heat capacity B) × (moulding temperature A / moulding temperature B)

= plastics rate of material A (kg/hr) × (QA/QB)

Q = total heat content of plastic (J/kg)

A = polystyrene

B = material actually to be used

Plasticizing rate = mass of moulding (kg) × number of moulding per hour

If the plasticizing is too low in relation to the shot size required, the chances are that the injected plastic will not yet be completely molten, whereas too high a capacity due to excessively long barrel dwell times.

Plasticizing vs shot size

Selection of the machine screw size usually depends only on the maximum shot size, but the plasticating ability can also be important. It is usually incorrect to assume that the screw's plasticating ability remains the same regardless of the shot size being used. As an example, when the screw reciprocates in preparing the melt, that may be 25 or 90% of shot capacity; thus, a portion of the screw feed section loses its ability to influences plastication.

2.8.3 Cycle time

The sequence of events during the injection mould of a plastic part is called the injection molding cycle. The cycle begins when the mould closes, followed by the injection of the polymer into the mould cavity. Once the cavity is filled, a holding pressure is maintained to compensate for material shrinkage. In the next step, the screw turns, feeding the next shot to the front screw. This causes the screw to retract as the next shot is prepared. Once the part is sufficiently cool, the mould opens and the part is ejected.

The formula of cycle time calculation:

$$\tau_c = (m \times 3600) / p$$

τ_c = minimum cycle time attainable (s)

m = mass of shot (kg)

p = plasticizing capacity of the machine with the polymer being moulded (kg/hr)

2.8.4 Clamping force

Clamping forces is one of the most commonly- used parameters for injection moulding machine. It refers to the final clamping force applied by the mould plate to the mould after the clamping unit has locked the mould and when the fused materials are being injected into the cavity. The clamping forces to a great extent reflect the ability of the injection moulding machine in processing product. Thus, most manufacturers take it as a parameter indicating the specification of injection moulding machine. When fused materials flow into the empty cavity with certain injection pressure and velocity, the clamping forces should satisfy the following formula to avoid expansion of the moulds by fused material. Insufficient clamping forces shall result in overflow:

The formula of clamping force calculation: **$F \geq aPA$**

Where: F = clamping forces (t)

a = safety coefficient, generally taking 1.1~1.6

P = cavity pressure (Mpa)

A = projection area of the mold parts on the parting line (cm^2)

2.8.5 The number of cavities

A number of factors should be considered when deciding how many cavities can be filled from a runner system. The number of cavities filled from a particular runner system can affect the ease with which the program arrives at a runner balancing solution. The possible combinations of pressure drops in the runner system itself and in filling the cavity volumes can rapidly increase the complexity of the balancing problem.

The number of cavities which can be filled from any given runner system depends on the following:

- Machine size (available clamp force)
- Available shot volume
- Available production time
- Required product quantity
- Shape and size of the moldings

a. Determination of cavity number by **Shot capacity:**

The injection machine shot capacity is also a factor in determining the number of cavities. Take 80 percent of the machine capacity as the shot weight (*S*) and divide by the part weight (*W*) to get the number of cavities. The relation is:

The formula of number of cavities by shot capacity= S / W

Where, S = 80% of machine capacity

W = part weight

b. Determination of cavity number by **Plasticizing capacity:**

The injection machine plasticizing capacity is also a factor. Divide the plasticizing capacity (*P*) of the machine by the estimated number of shots per minute (*X*) and part weight (*W*). The relation is:

The formula of the number of cavities by plasticizing capacity = **P /XW**

Where, P = machine plasticizing capacity

X = estimated number of shots per minute

W = part weight

c. Determination of cavity number by **Clamping capacity:**

Refer to the clamping capacity of the injection moulding. The formula is Rated clamping capacity (C) divided with by the estimated capacity pressure (MPa) approximately 63Mpa (Pc) and Projected area of moulding including runner and sprue (Am).

The formula of the number of number of cavities by Clamping capacity =**C / (Pc×Am)**

Where, C = Rated clamping capacity

Pc = capacity pressure (MPa) approximately 63Mpa

Am = Projected area of moulding including runner and sprue

2.8.6 Gate size calculation

Most part designers consider gate design to fall solely under the responsibility of the mold shop, and are therefore uninterested in gate design. This is short sighted. Poor gate design can reduce part performance and create cosmetics defects. Volumetric flow rate and gate size control shear rate in the gate. Excessive gate shear can lead to haziness around the gate (gate blush) and streaky defects.

a) The equation of depth of the gate, $h = nt$
b) The equation of gate width , $w = (n A^{1/2})/30$, $A = \pi d^2/4$

Where,

h = depth of the gate

t = component wall thickness

n = material constant (0.6)

w = gate width

A = surface area of cavity

2.8.7 Runner size calculation

The diameter of a runner highly depends on its length in addition to the part volume, part flow length, machine capacity, and gate size. Generally they must never be smaller than the largest wall thickness of the product and usually lay within the range 3 mm to 15 mm. recommended runner dimensions are provided in the table 1 below. The selection of a cold runner diameter should be based on standard machine tool cutter sizes.

Resin name	Runner diameter (mm)
ABS, AS	4.8 ~ 9.5
ACRYLATE (impact resistance)	8 ~ 9.5
ACRYLATE	8 ~ 12.7
POLYAMIDE 6	1.6 ~ 9.5
POLYCARBONATE	4.8 ~ 9.5
POLYETHYLENE	1.6 ~ 9.5
PPO	6.4 ~ 9.5
POLYSTYRENE	3.2 ~ 9.5
PVC	3.2 ~ 9.5

Table 1:Runner diameter based on type of resin

Runner size: $D = (W^{1/2} \times L^{1/4}) / 3.7$

Where : D = runner diameter (mm)

W = part weight (g)

L = runner length (mm)

2.9 Software

The software that used to generate the mould calculation software are visual basic and C++ programming.

2.9.1 Visual Basic

What is Visual Basic?

Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - **GUI**) applications. The applications have a familiar appearance to the user.

Visual Basic is **event-driven** (figure 2.15), meaning code remains idle until called upon to respond to some event (button pressing, menu selection,). Visual Basic is governed by an event processor. Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor [5].

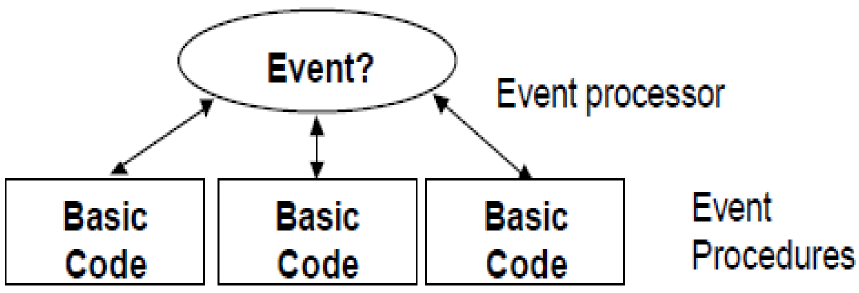


Figure 2.15: Event processor of visual basic

Some Features of Visual Basic:

- Full set of objects - you 'draw' the application
- Lots of icons and pictures for your use
- Response to mouse and keyboard actions
- Clipboard and printer access
- Full array of mathematical, string handling, and graphics functions
- Can handle fixed and dynamic variable and control arrays
- Sequential and random access file support
- Useful debugger and error-handling facilities
- Powerful database access tools
- ActiveX support
- Package & Deployment Wizard mark

2.9.2 C++ Programming

C++ (pronounced "see plus plus") is a statically typed, free-form, multi-paradigm, compiled, general-purpose programming language. It is regarded as an intermediate-level language, as it comprises a combination of both high-level and low-level language features. C++ is also used for hardware design, where the design is initially described in C++, then analyzed, architecturally constrained, and scheduled to create a register-transfer level hardware description language via high-level synthesis[10].

C++ is one of the most popular programming languages and is implemented on a wide variety of hardware and operating system platforms. As an efficient compiler to native code, its application domains include systems software, application software, device drivers, embedded software, high-performance server and client applications, and entertainment software such as video games. C++ has greatly influenced many other popular programming languages, most notably C# and Java.

Existing System

Based on Chan et al. (2003), Computerized Quotation System for Injection Mould Manufacturing (CQSIM) is one of many software systems that available in nowadays. The system is established with Microsoft Access and its programming language is Visual Basic for application. The Visual Basic application program calculate input data with predefine data from the database and save the calculated data into the corresponding result data table. The user must input data in this system such as mould base, ejector pin, cooling line and machining method to mould makers. Users choose a suitable operation from the menu for the desired tasks. The results can be obtained within a minute of calculation. Gumstix computer is a tiny single board computer which is an inexpensive and high-performance miniaturized platform to estimate the manufacturing parameter of moulded parts. It consists of a knowledge-base, knowledge processing units and server service unit for user interactions, all of which are implemented on the Gumstix computer.

CHAPTER 3

METHODOLOGY

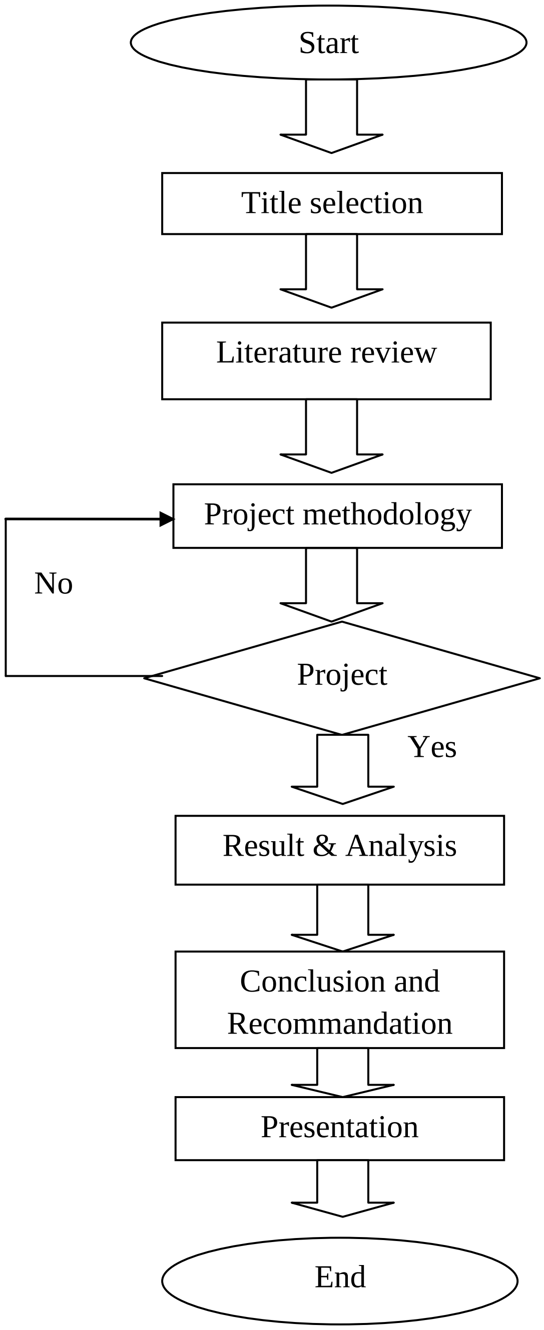
3.1 INTRODUCTION

This chapter introduces the overall methodological approach that is used in the project. In this project, several methods were used to complete the project and it will comprise all the methods and processes that will use in order to achieve our objectives which are to design mould parameter software system. To started this project with finding all material that related with project title for study and research such as journal, article and books.

After the research and study have been made, it followed with project planning. In this process have made a prediction and determine for the next process that needs to take in order to smooth the project progress. Through this chapter also, the full explanation for all process and tool that involved to experimental design To design mould parameter software system

Below are the process flow chart for implement the project and flow chart for the methodology processes which focus on design mould parameter software system

3.2 FLOW CHART OF METHODOLOGY



3.3 PROCESS FLOW CHART

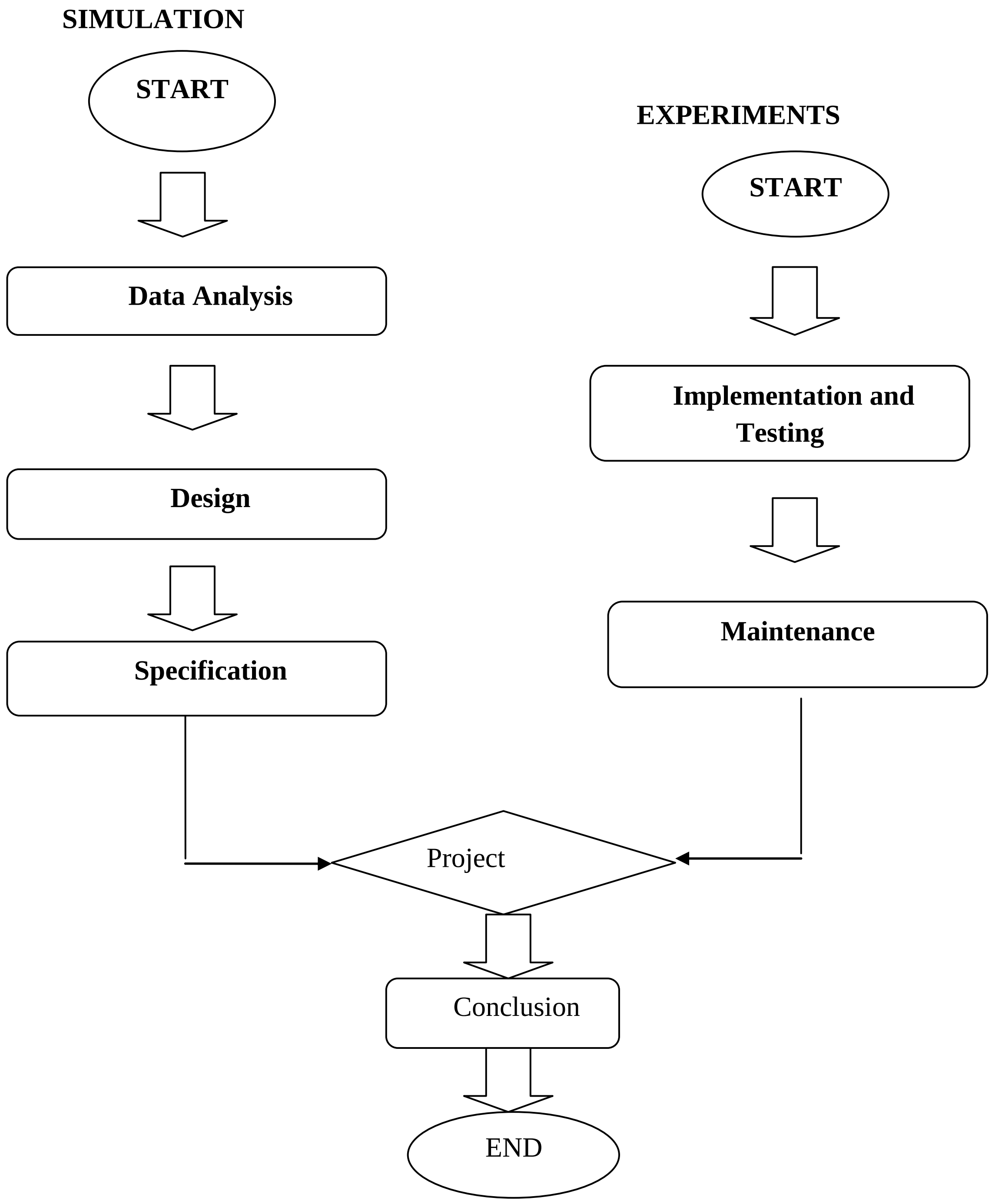


Figure 3.1: Process flow chart of simulation and experimental mould calculation software

3.4 Data Analysis

All part of analysis element will be conduct such as requirement analysis and planning analysis. Besides that, understanding of objectives, functions, requirements and planning of the project are very important to produces successfully project. The research are done by refer to books, read articles and journals from internet and discuss with industrial people who involve with this mould calculation software. Throughout this research, it is clear and understands the objectives of this project. There are two main objectives that is developing software There is another objective in developing this software system that is to develop mould parameter software system with the injection database.

3.5 Specification

System specification is produced from the detailed definition of the requirement analysis. This document should clearly define the product's function. This software system will calculate the parameter for plastic injection mould which is including calculation for shot capacity for plunger type and screw type, plasticizing capacity, cycle time, clamping force, number of cavity, gate and runner size.

3.6 Design

During this phase, the system specifications are translated into a software representation. The software at this stage is concern with the data structure of the system, software architecture of the system, algorithmic detail and interface representation of the system. The hardware requirements are also determined at this stage along with a picture of the overall system architecture. By the end of this stage, there were should be able to identify the relationship between the hardware, software and the associated interfaces. The system architecture for this software system is starting from the software developer where the developer develops the software with all the function that required in requirement stage. That software will be packaged into software package and install it into the computer. After the installation, the software system can successfully run through computer. The hardware requirements have been choosing after discuss with lecturer and related to the new project where a computer is needed to develop this project. Software requirements are using the Visual Basic application program, C++ programming and Microsoft Access 2007.

3.7 Interface of the Software

This paragraph will explain the suggestion design of the interface. It will show the interface of the mould calculation window. There are eight of the operations in the software.

3.7.1 Machine

From the main process (Figure 3.2), the user needs to select the type of the machine. The interface of the type of the machine will show.

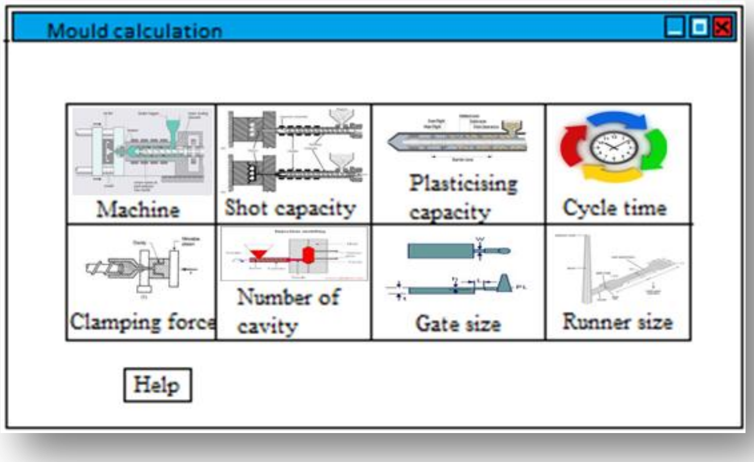


Figure 3.2: Main window

There are two types of the machine plunger type and screw type as shown in Figure 3.3. The user need to choose the type of the machine that user want to use.

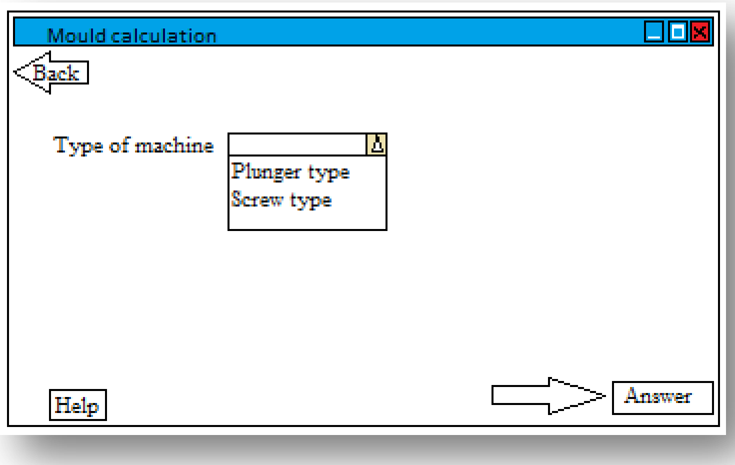


Figure 3.3: Type of machine window

3.7.2 Shot capacity

From the main window (Figure 3.4), the user needs to select the shot capacity process. The next interface will show.

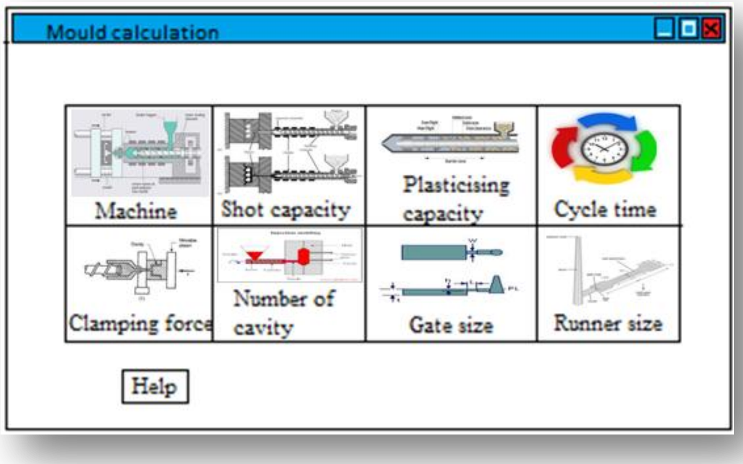


Figure 3.4: Main window

This interface (Figure 3.5) shows the shot capacity calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

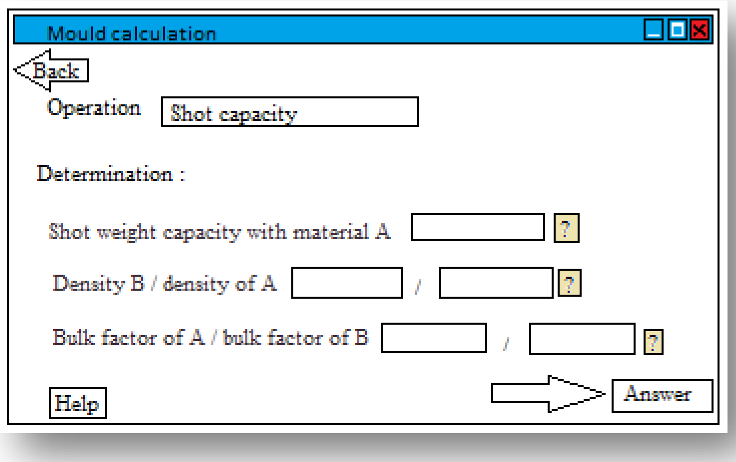


Figure 3.5: Shot capacity calculation window

3.7.3 Plasticizing capacity

From the main window (Figure 3.4), the user needs to select the plasticizing capacity process. The next interface will show.

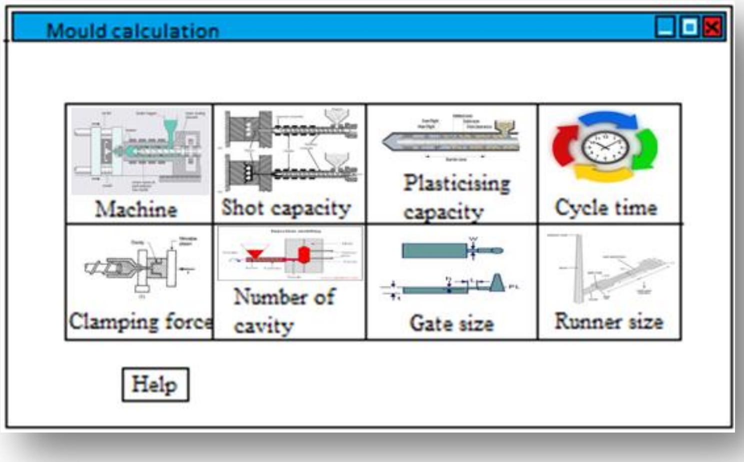


Figure 3.6: Main window

This interface (Figure 3.7) shows the plasticizing capacity calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

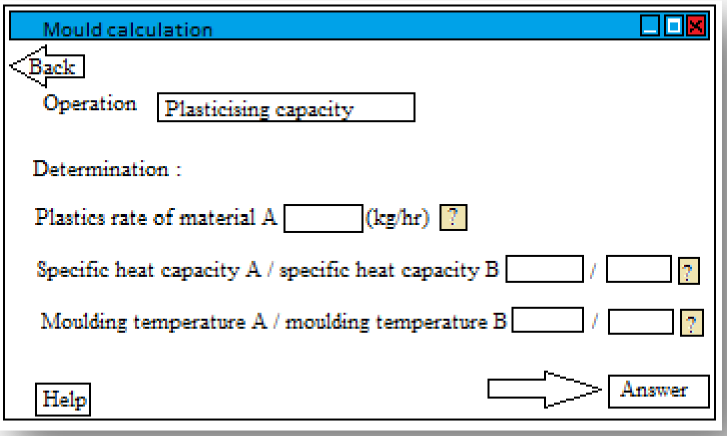


Figure 3.7: Plasticizing capacity calculation window

3.7.4 Cycle time

From the main window (Figure 3.8), the user needs to select the cycle time process. The next interface will show.

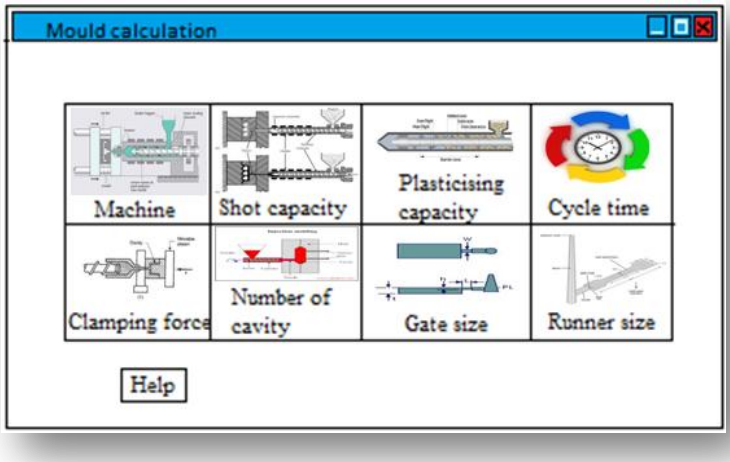


Figure 3.8: Main window

This interface (Figure 3.9) shows the cycle time calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

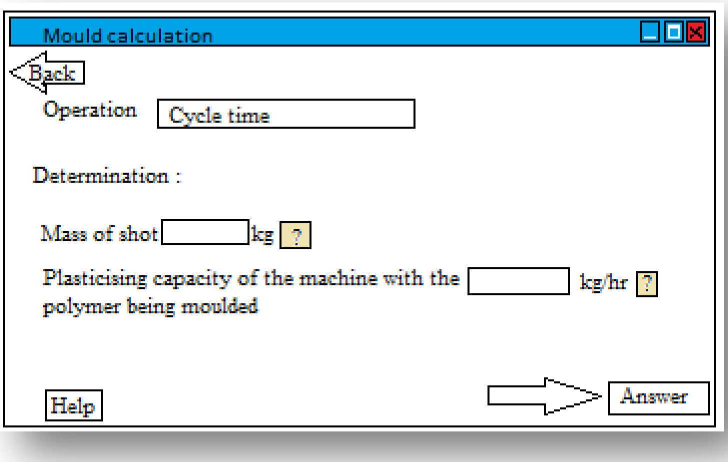


Figure 3.9: Cycle time calculation window

3.7.5 Clamping force

From the main window (Figure 3.10), the user needs to select the clamping force process. The next interface will show.

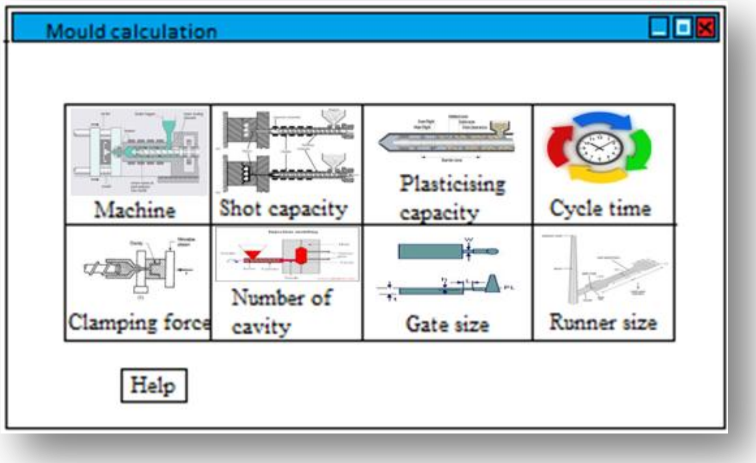


Figure 3.10: Main window

This interface (Figure 3.11) shows the clamping force calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

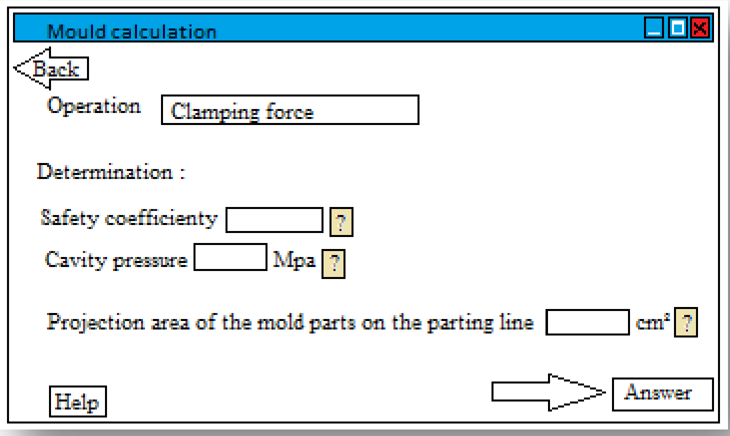


Figure 3.11: Clamping force calculation window

3.7.6 Number of cavity

From the main window (Figure 3.12), the user needs to select the number of cavity process. The next interface will show.

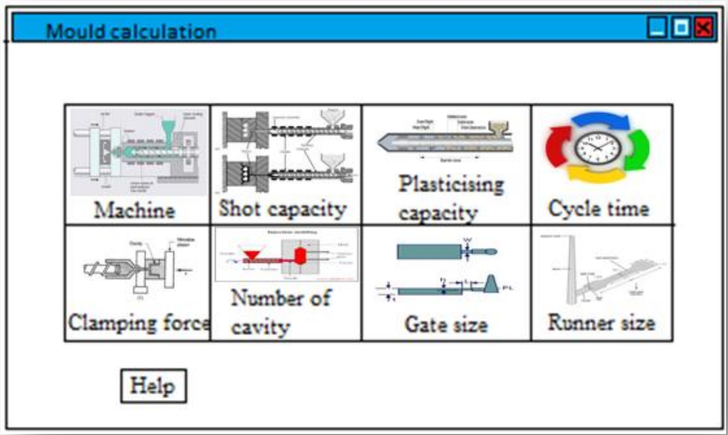


Figure 3.12: Main window

After select the number of cavity calculation, the user needs to select the operation of the number of cavity. There are three operations for number of cavity calculation which is by shot capacity, by plasticizing capacity and by clamping capacity as shown in Figure3.13.

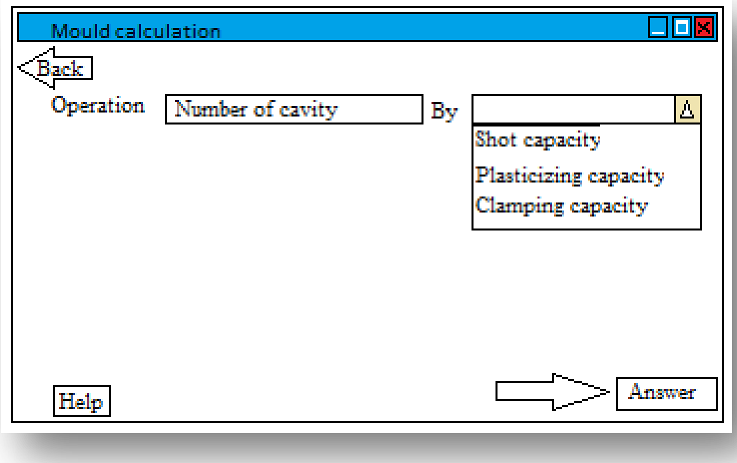


Figure 3.13: Type of number of cavity window

3.7.6.1 Shot capacity

This interface (Figure 3.14) shows the number of cavity by shot capacity calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

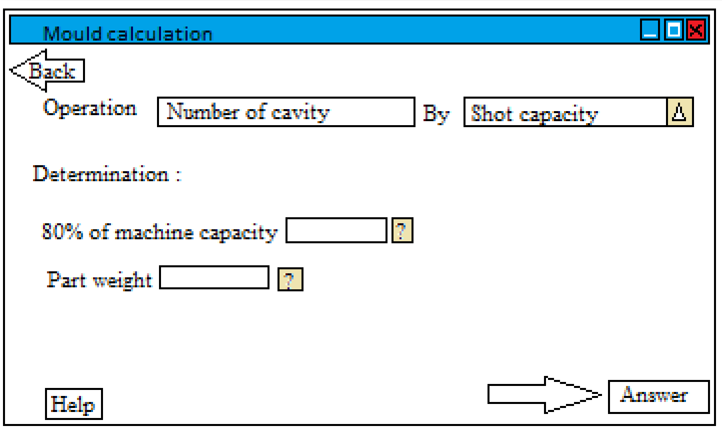


Figure 3.14: Clamping force calculation window

3.7.6.2 Plasticizing capacity

This interface (Figure 3.15) shows the number of cavity by plasticizing capacity calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button

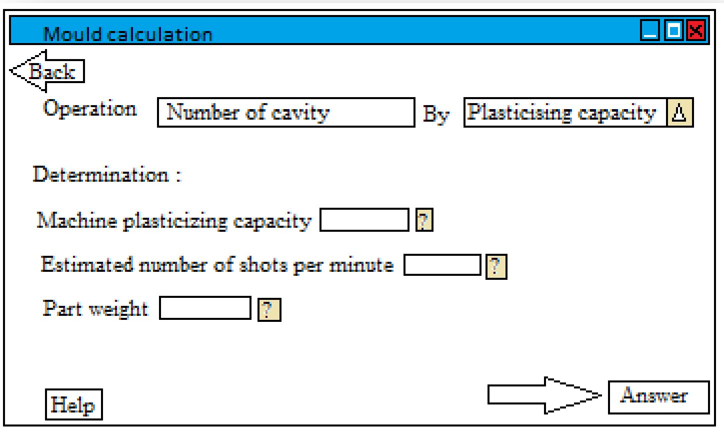


Figure 3.15: Plasticizing capacity calculation window

3.7.6.3 Clamping capacity

This interface (Figure 3.16) shows the number of cavity by clamping capacity calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button

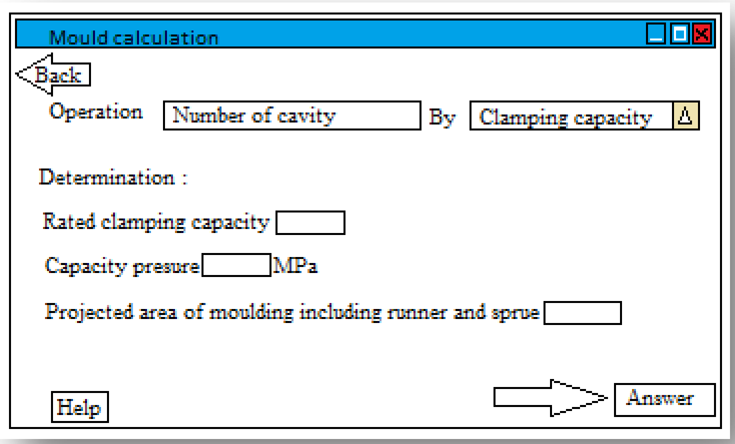


Figure 3.16: Clamping capacity calculation window

3.7.7 Gate size

From the main window (Figure 3.17), the user needs to select the gate size process. The next interface will show.

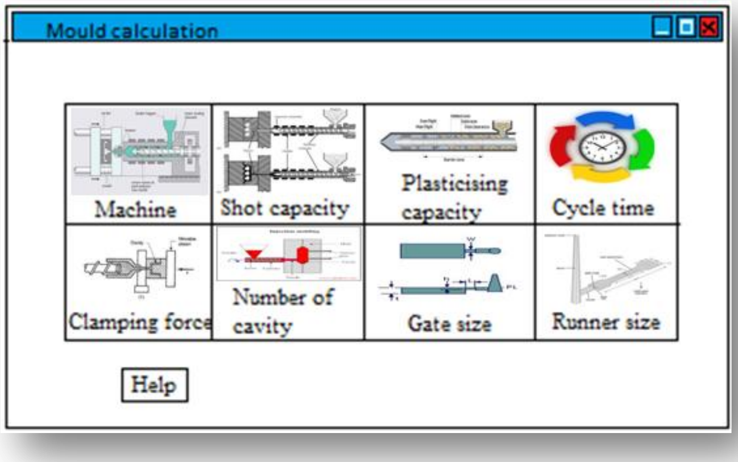


Figure 3.17: Main window

After select the gate size calculation, the user needs to select the operation of the gate size. There are two operations for gate size calculation which is by depth and by width as shown in Figure 3.18.

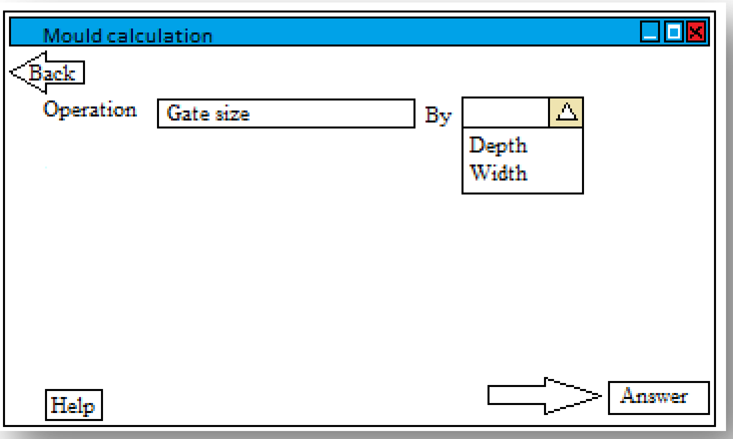


Figure 3.18: Type of gate size calculation window

3.7.7.1 Depth

This interface (Figure 3.19) shows the gate size by depth calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button

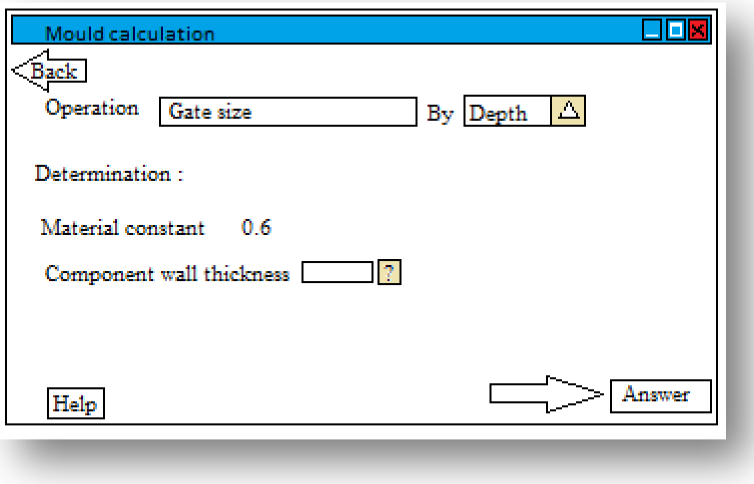


Figure 3.19: Gate size by depth calculation window

3.7.7.2 Width

This interface (Figure 3.20) shows the gate size by width calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button.

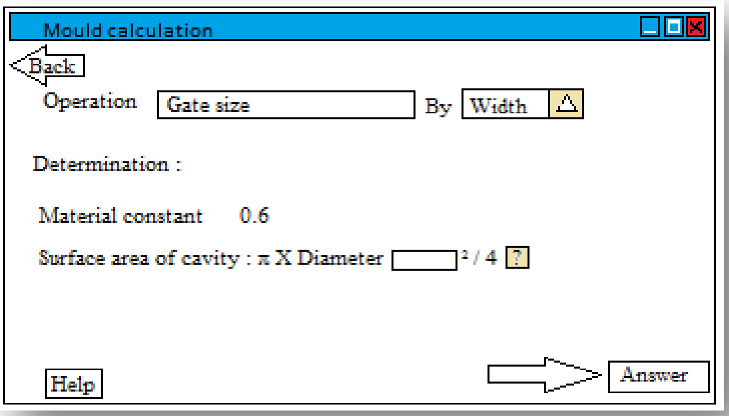


Figure 3.20: Gate size by width calculation window

3.7.8 Runner size

From the main window (Figure 3.21), the user needs to select the runner size process. The next interface will show.

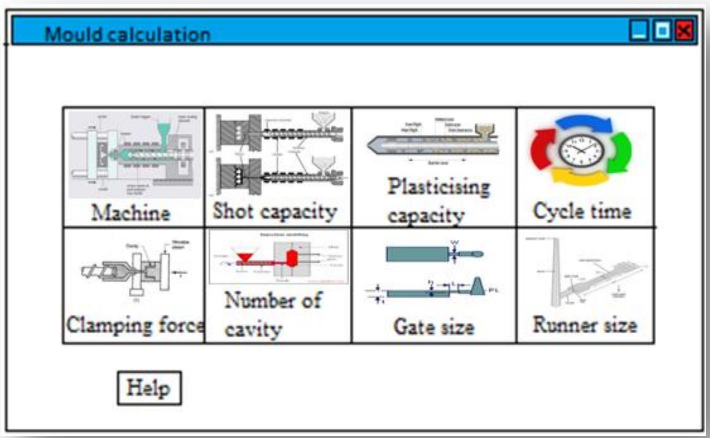


Figure 3.21: Main window

This interface (Figure 3.22) shows the runner size calculation. The user needs to fill in the parameter in the box. The answer will appear after the user click the answer button

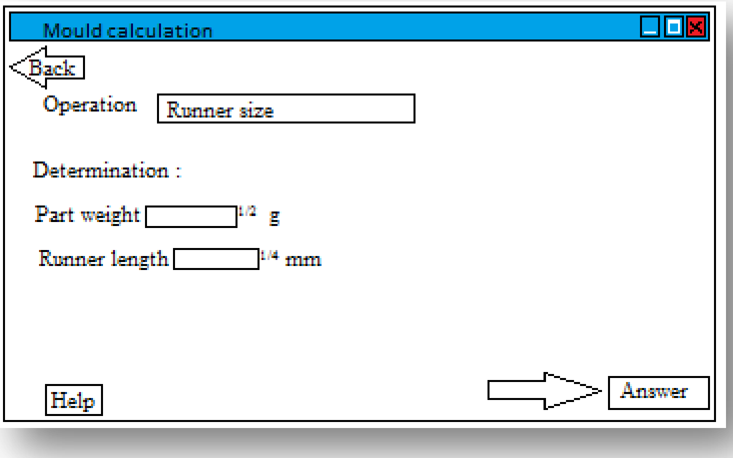


Figure 3.22: Runner size calculation window

3.8 Implementation and Testing

In this stage, the designs of the system will translated into the software domain. Detailed documentation from the design phase can significantly reduce the coding effort. The testing during this stage focuses on making sure that any errors are identified and that the software meets its required speciation. After the system are finished developed, it will do testing on the software system before make the system packaged. If it couldn't figure out any bugs in this software system, the software will be packaged into packaged software to be installed in computer. It will & a testing again after the installation the software system in computer to figure out any bugs in the system before the system go to next step. It will troubleshoot the system after the system installed in the computer to avoid any errors or bugs in the system the customer do the user acceptance test in next stage (maintenance).

3.9 Maintenance

This phase is the usually the longest stage of the software. In this phase the software is updated to few things such as meet the changing specification needs and adapted to accommodate changes in the external environment. Besides that, this phase also update error correction and oversights previously undetected in the testing phases and enhancing the efficiency of the software. Observe that feedback loops allow for correction to be incorporated into the model. When changes are made at any phase, the relevant documentation should be updated to reflect that change. The maintenance for this software system is updating the latest update of the system such as developer widens the scope of this software system.

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

This chapter is shows a result from the development of mould calculation software and experimental setup. This software system will calculate the parameter for plastic injection mould which is including calculation for **shot capacity** for plunger type and screw type**, plasticizing capacity**, **cycle time**, **clamping force**, **number of cavity, gate and runner size**. The visual basic software is used to simulate the mould calculation interface.

4.2 THE SOFTWARE

This section discuss about the coding and the interface of the software that had been designed. It shows the process of the software being design and the flow of the software

4.2.1 Front window-Plunger Type

The front window is designed to have the features for selecting the type of the machine. One of the machines is plunger type. Hence, it is designed to calculate shot capacity, plasticizing capacity, clamping force, cycle time, number of cavity, gate and runner size. For the plunger type machine, in order to achieve the window design, the interface is designed as shown in Figure 4.1 and consists of plunger type button.

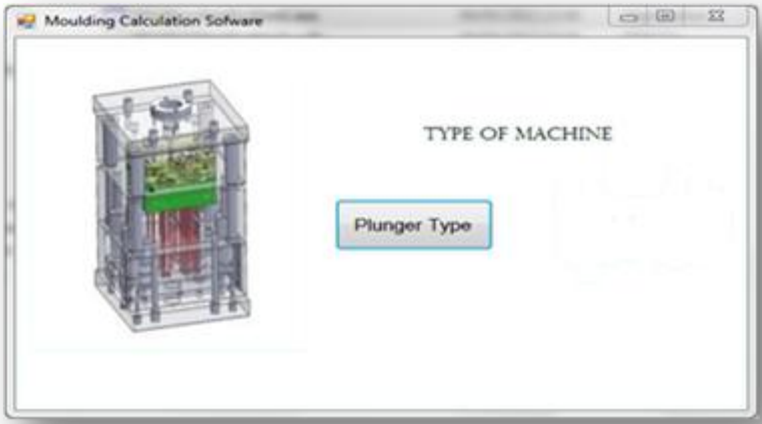


Figure 4.1: Type of the machine-Plunger Type

4.2.2 Shot Capacity Calculation

The shot capacity calculation is input into the program. The program is then generated as shown in Figure 4.2.

```
PublicClass Form1

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form2.Show()
Me.Hide()

EndSub

EndClass
```

Figure 4.2: The coding of the plunger type window

After the code is generated, the next window is created as shown in Figure 4.3. It shows the shot capacity icon in the window when referring to the code "*Form2.show*".

Figure 4.3: Plunger Type window

To develop the shot capacity calculation, the parameter is input into the program. The parameters are shot weight capacity with material A, density b / density of A, and bulk factor of A / bulk factor of B. The program is then generated as show in Figure 4.4.

```
PublicClass Form3

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger
Dim c AsInteger
Dim d AsInteger
Dim f AsInteger

        a = TextBox1.Text
        b = TextBox2.Text
        c = TextBox3.Text
        d = TextBox4.Text
        f = TextBox5.Text

        TextBox6.Text = a * (b / c) * (d / f)


EndSub
EndClass
```

Figure 4.4: The coding of shot capacity calculation

After the code is generated, the shot capacity window is generated as show in Figure 4.5. It consists of empty columns where users will key in the data for shot capacity calculation. When click answer button, calculation will refer to the program: *TextBox6.Text = a * (b / c) * (d / f)*. The result of shot capacity is generated automatically.

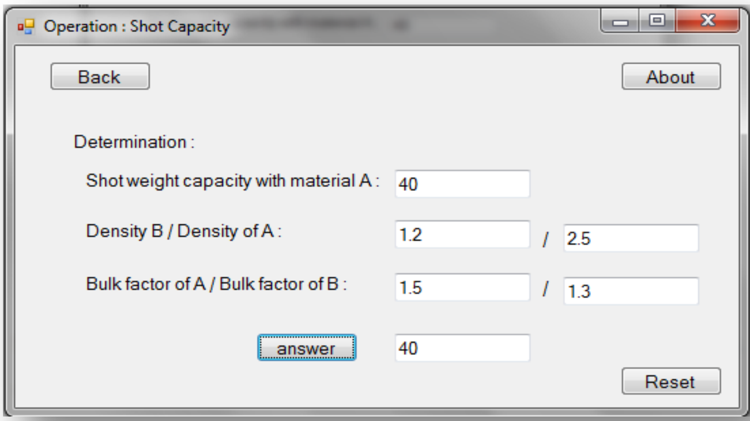


Figure 4.5: The shot capacity window

4.2.2.1 About window – Shot Capacity Calculation

The About window is created to help the user to referring the definition and formula of the shot capacity. The program is then generated as shown in Figure 4.6.

```
PublicClass Form3

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form4.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.6: The coding of About window

After the code is generated and the About window is created as show in Figure 4.7 by referring to the code "*Form4.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.7.

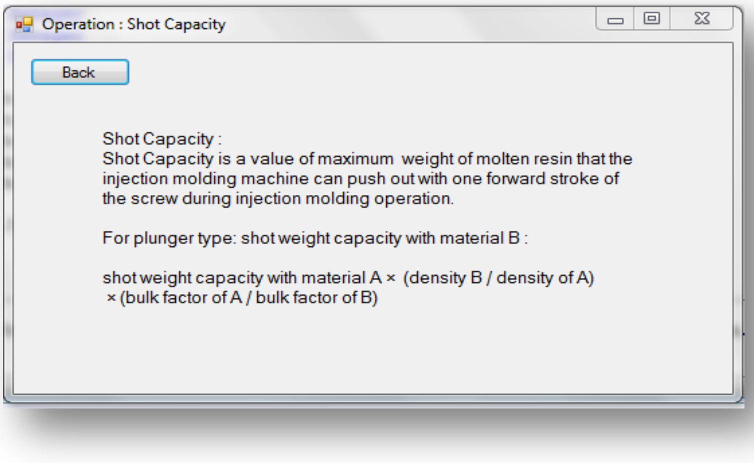


Figure 4.7: The About window

4.2.3 The Plasticizing Capacity Calculation - The Plunger and the Screw Type

The plasticizing capacity calculation is input into the program. The program is then generated as shown in Figure 4.8.

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()

EndSub

EndClass
```

Figure 4.8: The coding of plasticizing capacity window

After the code is generated, the next window is created as shown in Figure 4.9. It shows the plasticizing capacity calculation icon in the window when referring to the code "*Form5.show*".

Figure 4.9: The process operation plasticizing capacity window

In order to develop the plasticizing capacity calculation, the parameters are input into the program. The parameter are plastics rate of material A, specific heat capacity A / specific heat capacity B and moulding temperature A / moulding temperature B. The program is then generated as show in Figure 4.10.

```
PublicClass Form5

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim g AsInteger
Dim h AsInteger
Dim i AsInteger
Dim j AsInteger
Dim k AsInteger

        g = TextBox1.Text
        h = TextBox2.Text
        i = TextBox3.Text
        j = TextBox4.Text
        k = TextBox5.Text

        TextBox6.Text = g * (h / i) * (j / k)
EndSub
EndClass
```

Figure 4.10: The coding of plasticizing capacity calculation

After the code is generated, the plasticizing capacity window is generated as show in Figure 4.11. It consists of empty columns where users will key in the data for plasticizing capacity calculation. When click answer button, calculation will refer to the program: *TextBox6.Text = g * (h / i) * (j / k)*. The result of shot capacity is generated automatically.

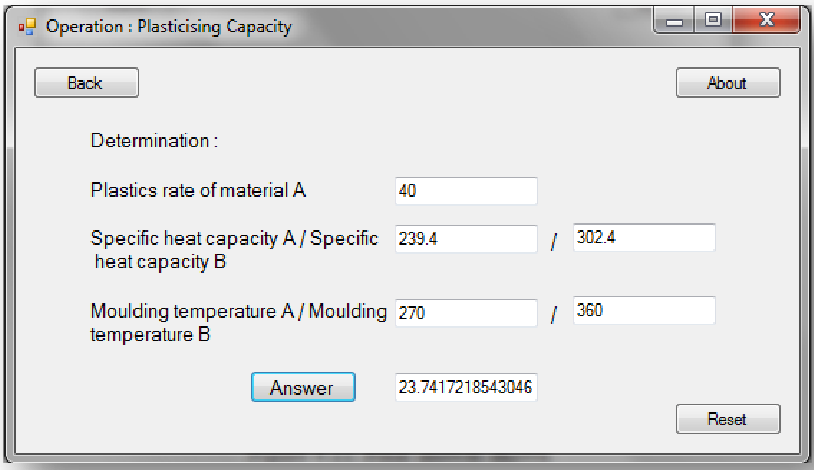


Figure 4.11: The plasticizing capacity window

4.2.3.1 About window – Plasticizing Capacity Calculation

The About window is created to help the user to referring the definition and formula of the plasticizing capacity. The program is then generated as shown in Figure 4.12.

```
PublicClass Form5

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form6.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.12: The coding About window

After the code is generated and the About window is created as show in Figure 4.13 by referring to the code "*Form6.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.13.

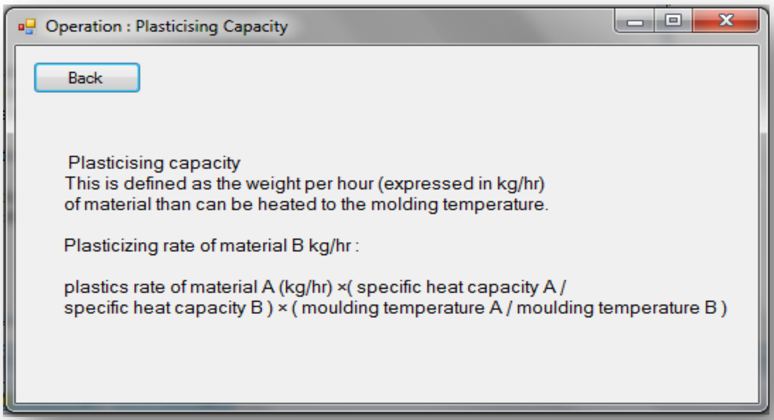


Figure 4.13: The About window

4.2.4 The Clamping Force Calculation - The Plunger and the Screw Type

For the clamping force calculation, the code for generate the window is input into the program. The program is then generated as shown in Figure 4.14.

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object,ByVal e As
System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object,ByVal e As
System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object,ByVal e As
System.EventArgs) Handles Button3.Click
        Form7.Show()
Me.Close()
EndSub

EndClass
```

Figure 4.14: The coding of clamping force window

After the code is generated, the next window is created as shown in Figure 4.15. It shows the clamping force calculation icon in the window when referring to the code "*Form7.show*".

Figure 4.15: The clamping force window

To develop the clamping force calculation, the parameter is input into the program. The parameters are safety co efficiency, cavity pressure and projection area of the mould parts on the parting line. The program is then generated as show in Figure 4.16.

```
PublicClass Form7

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger
Dim c AsInteger

        a = TextBox1.Text
        b = TextBox2.Text
        c = TextBox3.Text

        TextBox4.Text = a * b * c

EndSub
EndClass
```

Figure 4.16: The coding of clamping force calculation

After the code is generated, the clamping force window is generated as show in Figure 4.17. It consists of empty columns where users will key in the data for clamping force calculation. When click answer button, calculation will refer to the program: TextBox4.Text = a * b * c. The result of clamping force is generated automatically.

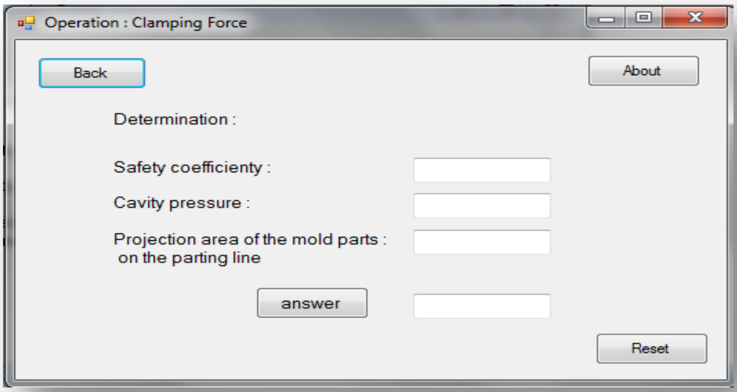


Figure 4.17: The clamping force window

4.2.4.1 About window – Clamping Force Calculation

The About window is created to help the user to referring the definition and formula of the clamping force. The program is then generated as shown in Figure 4.18.

```
PublicClass Form7

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form8.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.18: The coding of About window

After the code is generated and the About window is created as show in Figure 4.19 by referring to the code "*Form8.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.19.

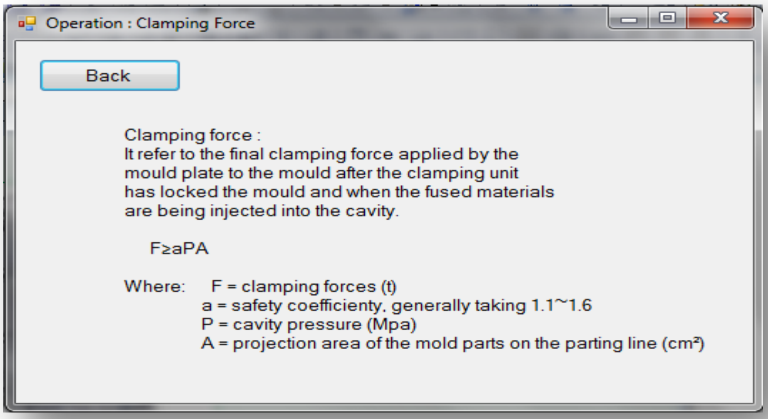


Figure 4.19: The About window

4.2.5 The Number of Cavity Calculation - The Plunger and the Screw Type

The number of cavity calculation is input into the program. The program is then generated as shown in Figure 4.20.

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Form7.Show()
Me.Close()
EndSub
PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Form9.Show()
Me.Close()
EndSub

EndClass
```

Figure 4.20: The coding of clamping force calculation

After the code is generated, the next window is created as shown in Figure 4.21. It shows the number of cavity calculation icon in the window when referring to the code "*Form9.show*".

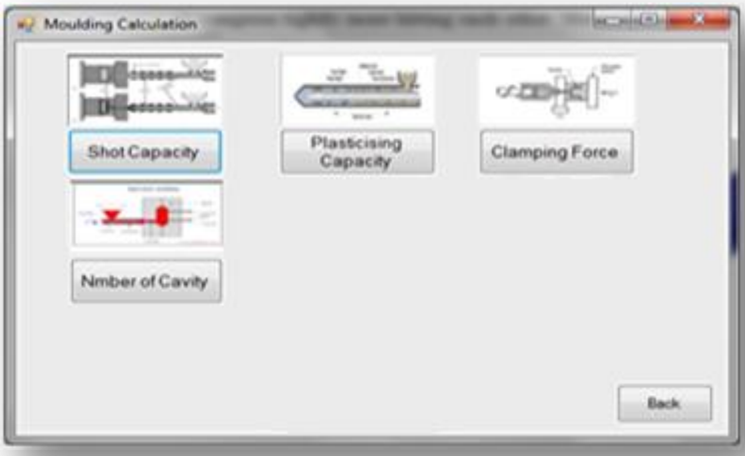


Figure 4.21: The number of cavity window

4.2.5.1 Shot Capacity

The type of number of cavity is input into the program. The number of cavity is by shot capacity. The program is then written as shown in Figure 4.22.

```
PublicClass Form9

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form10.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.22: The code of the number of cavity window

After the code is generated, the next window is created as shown in Figure 4.23. It shows the shot capacity icon in the window when referring to the code "*Form10.show*".

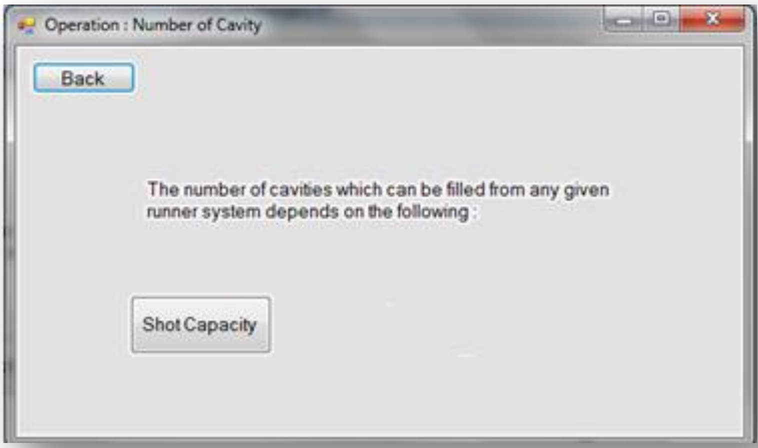


Figure 4.23: The number of cavity - shot capacity window

In order to achieve the number of cavity by shot capacity calculation, the parameter is input into the program. The parameters are 80% of machine capacity and part weight, the program is then generated as shown in Figure 4.24.

```
PublicClass Form10

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim a AsInteger
Dim b AsInteger

        a = TextBox1.Text
        b = TextBox2.Text

        TextBox3.Text = (a * 0.8) / b
EndSub
EndClass
```

Figure 4.24: The coding of number of cavity - shot capacity

After the code is generated, the number of cavity by shot capacity window is generated as show in Figure 4.25. It consists of empty columns where users will key in the data for number of cavity by shot capacity calculation. When click answer button, calculation will refer to the program: *TextBox3.Text = (a * 0.8) / b*. The result of number of cavity by shot capacity is generated automatically.

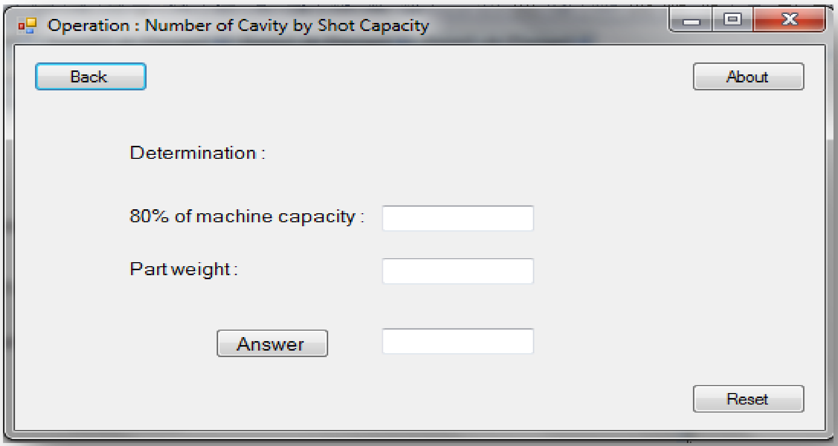


Figure 4.25: The number of cavity - shot capacity window

4.2.5.2 About window – Number of Cavity by Shot Capacity

The About window is created to help the user to referring the definition and formula of the number of cavity by shot capacity. The program is then generated as shown in Figure 4.26.

```
PublicClass Form10

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form11.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.26: The coding of About window

After the code is generated and the About window is created as show in Figure 4.27 by referring to the code "*Form11.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.27.

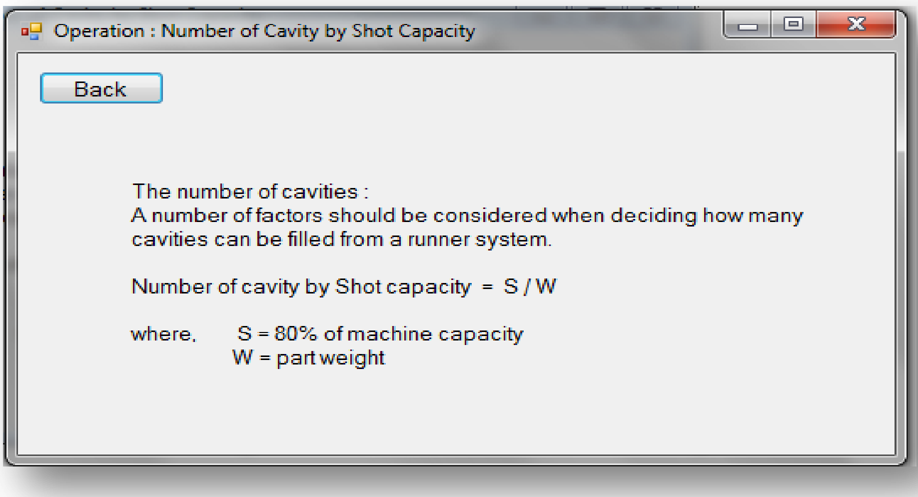


Figure 4.27: The About window

4.2.5.3 Plasticizing Capacity

The type of number of cavity is input into the program. The number of cavity calculation is by shot capacity. The program is then written as shown in Figure 4.28.

```
PublicClass Form9

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form10.Show()
Me.Close()

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form12.Show()
Me.Close()
EndSub
```

Figure 4.28: The code of the number of cavity window

After the code is generated, the next window is created as shown in Figure 4.29. It shows the plasticizing capacity icon in the window when referring to the code "*Form12.show*".

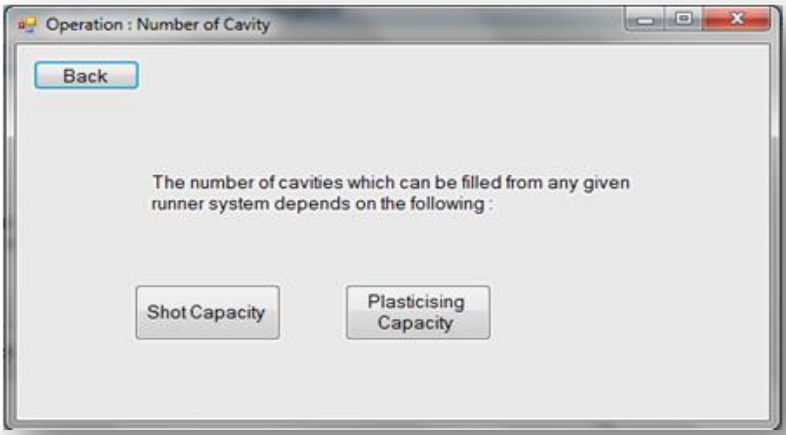


Figure 4.29: The number of cavity - plasticizing capacity window

To develop the number of cavity by plasticizing capacity calculation, the parameter is input into the program. The parameters are machine plasticizing capacity, estimated number of shots per minute and part weight, the program is then generated as shown in figure 4.30.

```
PublicClass Form12

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger
Dim c AsInteger

        a = TextBox1.Text
        b = TextBox2.Text
        c = TextBox3.Text

        TextBox4.Text = a / (b * c)
EndSub

EndClass
```

Figure 4.30: The coding of number of cavity - plasticizing capacity

After the code is generated, the number of cavity by plasticizing capacity window is generated as show in Figure 4.31. It consists of empty columns where users will key in the data for number of cavity by plasticizing capacity calculation. When click answer button, calculation will refer to the program: *TextBox4.Text = a / (b * c)*. The result of number of cavity by shot capacity is generated automatically.

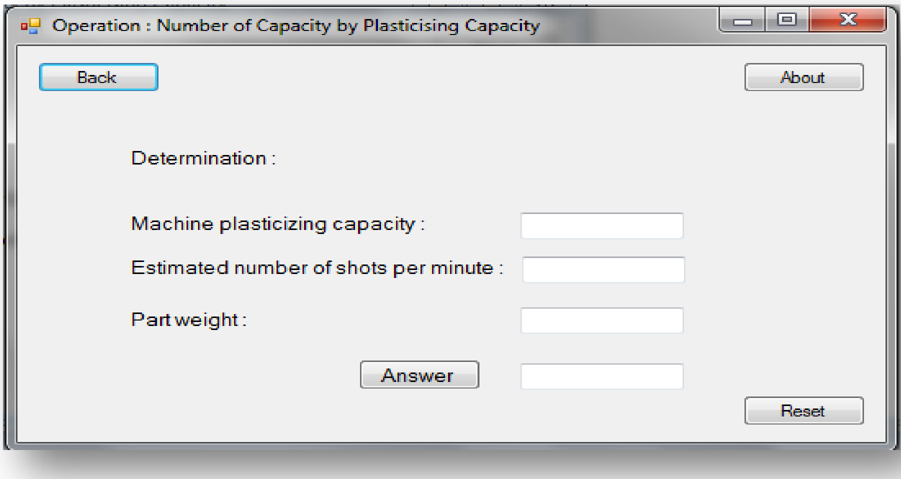


Figure 4.31: The number of cavity - plasticizing capacity window

4.2.5.4 About window - The Number of Cavity by Plasticizing Capacity

The About window is created to help the user to referring the definition and formula of the number of cavity by shot capacity. The program is then generated as shown in Figure 4.32.

```
PublicClass Form12

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form13.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.32: The coding of About window

After the code is generated and the About window is created as show in Figure 4.33 by referring to the code "*Form13.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.33.

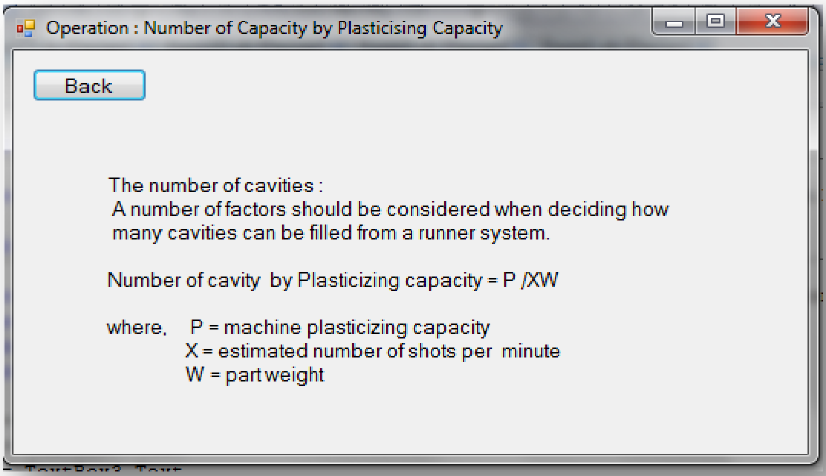


Figure 4.33: The About window

4.2.5.5 Clamping Force

In order to achieve the number of cavity calculation, the type of number of cavity is input into the program. The number of cavity calculation is by clamping force. The program is then written as shown in Figure 4.34.

```
PublicClass Form9

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form10.Show()
Me.Close()

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form12.Show()
Me.Close()

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Form14.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.34: The code of the number of cavity calculation

After the code is generated, the next window is created as shown in Figure 4.35. It shows the clamping force icon in the window when referring to the code "*Form14.show*"

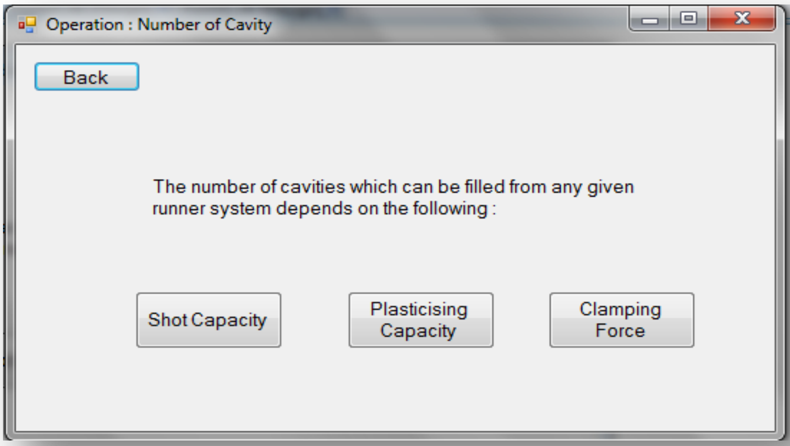


Figure 4.35: The number of cavity – Clamping force window

To develop the number of cavity by clamping force calculation, the parameter is input into the program. The parameters are rated clamping capacity, capacity pressure and projected area of moulding runner and sprue, the program is then generated as shown in Figure 4.36.

```
PublicClass Form14

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger
Dim c AsInteger

        a = TextBox1.Text
        b = TextBox2.Text
        c = TextBox3.Text

        TextBox4.Text = a / (b * c)
EndSub
EndClass
```

Figure 4.36: The coding of number of cavity - clamping force

After the code is generated, the number of cavity by clamping force window is generated as show in Figure 4.31. It consists of empty columns where users will key in the data for number of cavity by clamping force calculation. When click answer button, calculation will refer to the program: *TextBox4.Text = a / (b * c)*. The result of number of cavity by clamping force is generated automatically

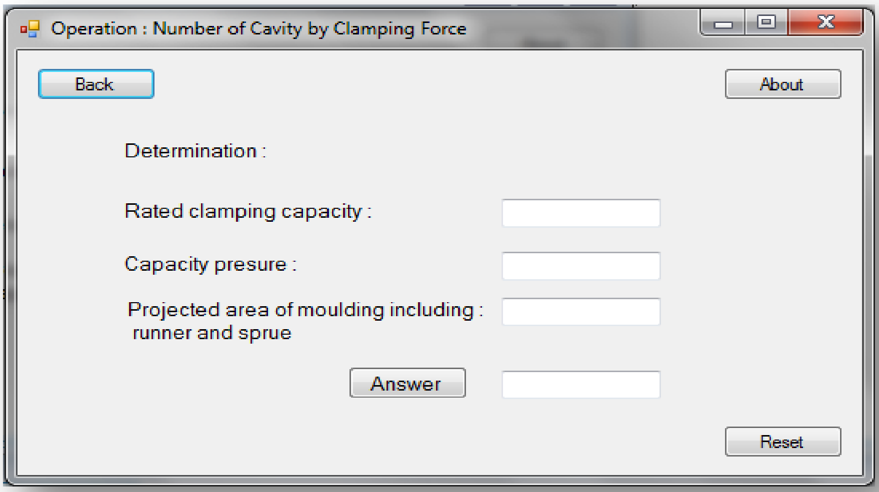


Figure 4.37: The number of cavity - clamping force window

4.2.5.6 About window - The Number of Cavity by Clamping Force

The About window is created to help the user to referring the definition and formula of the number of cavity by clamping force. The program is then generated as shown in Figure 4.38.

```
PublicClass Form14

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Form15.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.38: The coding of the number of cavity - shot capacity

After the code is generated and the About window is created as show in Figure 4.39 by referring to the code "*Form15.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.39.

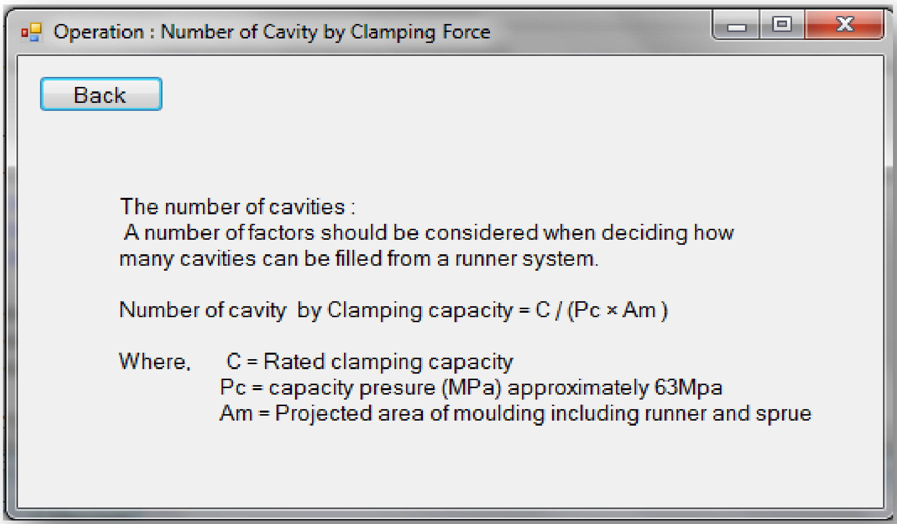


Figure 4.39: The About window

4.2.6 The Gate Size Calculation - The Plunger and the Screw Type

For the gate size calculation, the code for generate the window is input into the program. The program is then generated as shown in Figure 4.40

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()

EndSub

   PrivateSub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()

EndSub

  PrivateSub Button3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button3.Click
        Form9.Show()
Me.Close()

EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
        Form7.Show()
Me.Close()

EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button5.Click
        Form16Show()
Me.Close()
```

Figure 4.40: The coding of the gate size

After the code is generated, the next window is created as shown in Figure 4.41. It shows the gate size icon in the window when referring to the code "*Form16.show*".

Figure 4.41: The gate size window

4.2.6.1 The Depth

In order to develop the gate size calculation, the type of gate size is input into the program. The gate size calculation is by depth. The program is then written as shown in Figure 4.42.

```
PublicClass Form16

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form17.Show()
Me.Close()

EndSub

EndClass
```

Figure 4.42: The coding of the gate size - the depth calculation

After the code is generated, the next window is created as shown in Figure 4.43. It shows the depth icon in the window when referring to the code "*Form17.show*".

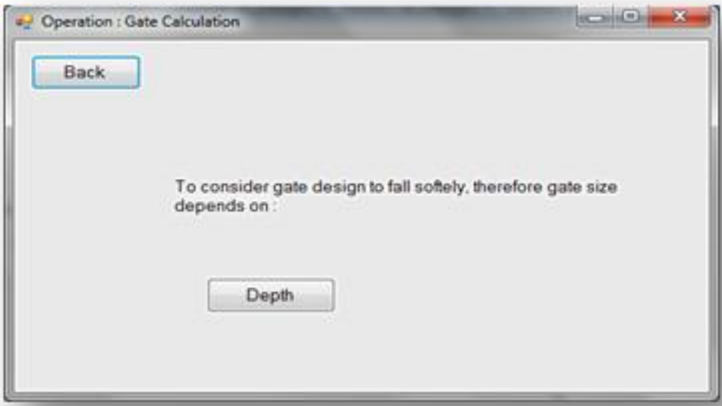


Figure 4.43: The gate size – depth window

To achieve the gate size by the depth calculation, the parameter is input into the program. The parameters are material constant (0.6) and component wall thickness, the program is then generated as shown in Figure 4.44.

```
PublicClass Form17

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim a AsInteger

        a = TextBox1.Text

        TextBox2.Text = 0.6 * a
EndSub

EndClass
```

Figure 4.44: The coding of the gate size -the depth

After the code is generated, the gate size by depth window is generated as show in Figure 4.45. It consists of empty columns where users will key in the data for the gate size by depth calculation. When click answer button, calculation will refer to the program: *TextBox2.Text = 0.6 * a* . The result of the gate size by depth is generated automatically.

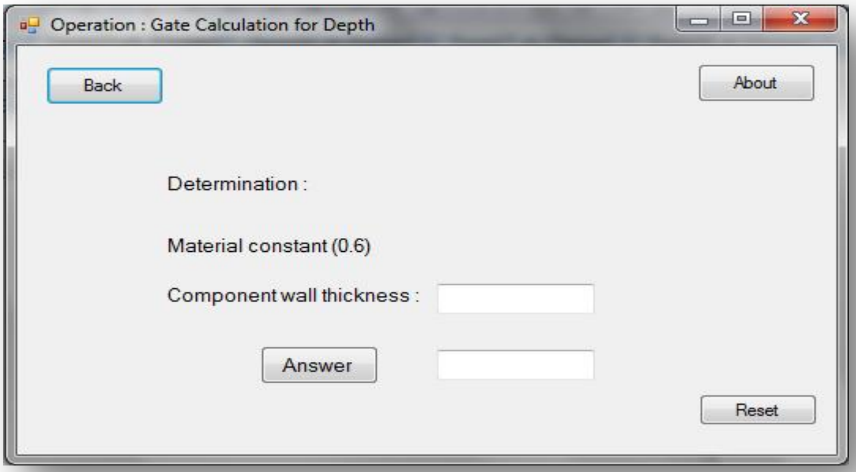


Figure 4.45: The gate size – depth window

4.2.6.2 About window - The Gate Size by Depth

The About window is created to help the user to referring the definition and formula of the gate size by depth. The program is then generated as shown in Figure 4.46.

```
PublicClass Form17

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form18.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.46: The coding gate size – depth

After the code is generated and the About window is created as show in Figure 4.47 by referring to the code "*Form18.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.47.

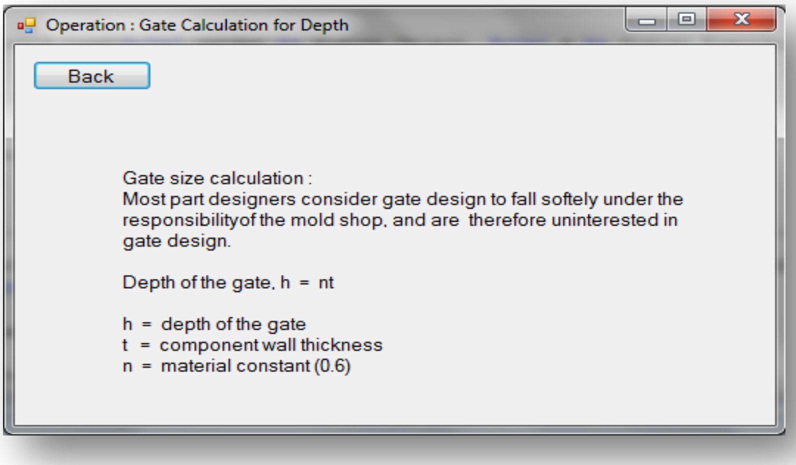


Figure 4.47: The About window

4.2.6.3 The Width

For the gate size calculation, the type of gate size is input into the program. The gate size calculation is by width. The program is then written as shown in Figure 4.48.

```
PublicClass Form16

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form17.Show()
Me.Close()

EndSub
`
PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Form19.Show()
Me.Close()

EndClass
```

Figure 4.48: The coding of the gate size - the width

After the code is generated, the next window is created as shown in Figure 4.49. It shows the width icon in the window when referring to the code "*Form19.show*".

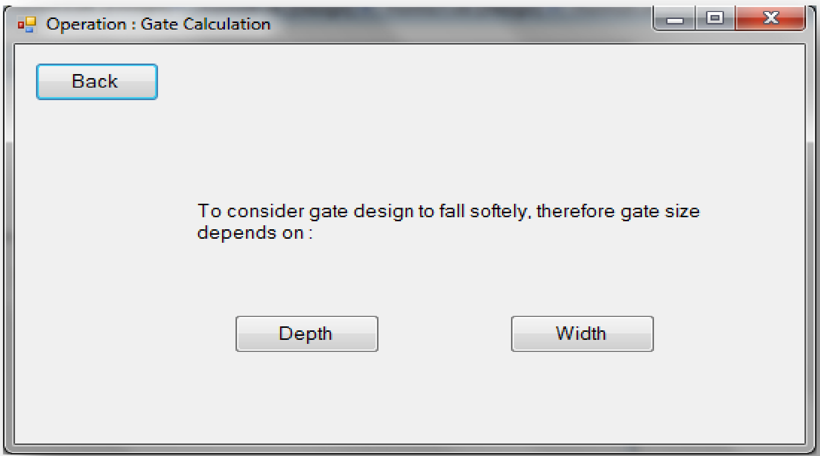


Figure 4.49: The gate size – width window

To develop the gate size by the width calculation, the parameter is input into the program. The parameters are material constant (0.6) and surface area of cavity, A: diameter, the program is then generated as shown in Figure 4.50.

```
PublicClass Form19

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim a AsInteger

        a = TextBox1.Text

        TextBox2.Text = (0.6 * ((3.145 * a ^ 2) / 4) ^ 1 / 2) / 30
EndSub

EndClass
```

Figure 4.50: The coding of gate size – width

After the code is generated, the gate size by width window is generated as show in Figure 4.51. It consists of empty columns where users will key in the data for the gate size by width calculation. When click answer button, calculation will refer to the program: *TextBox2.Text = (0.6 * ((3.145 * a ^ 2) / 4) ^ 1 / 2) / 30* . The result of the gate size by width is generated automatically.

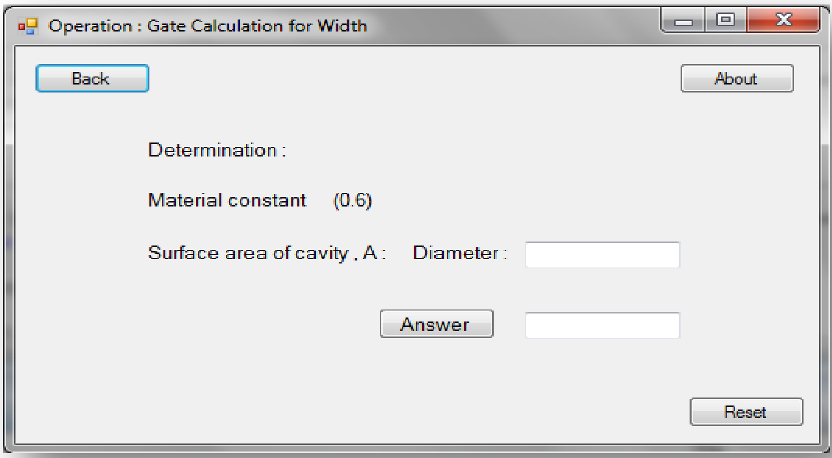


Figure 4.51: The gate size – width window

4.2.6.4 About window - The Gate Size by Width

The About window is created to help the user to referring the definition and formula of the gate size by width. The program is then generated as shown in Figure 4.52.

```
PublicClass Form19

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form20.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.52: The coding gate size – depth

After the code is generated and the About window is created as show in Figure 4.53 by referring to the code "*Form20.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.53.

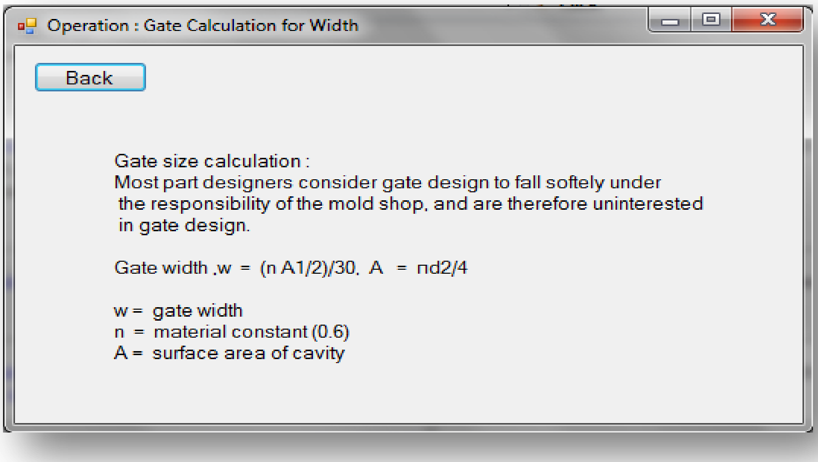


Figure 4.53: The About window

4.2.7 The Runner Size Calculation - The Plunger and the Screw Type

The runner size calculation is input into the program. The program is then generated as shown in Figure 4.54.

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Form9.Show()
Me.Close()

EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Form7.Show()
Me.Close()

EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Form16Show()
Me.Close()

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
        Form21.Show()
Me.Close()

EndSub

EndClass
```

Figure 4.54: The coding of the runner size

After the code is generated, the next window is created as shown in Figure 4.55. It shows the runner size icon in the window when referring to the code "*Form21.show*".

Figure 4.55: The runner size window

In order to achieve the runner size calculation, the parameter is input into the program. The parameters are part weight and runner length, the program is then generated as shown in Figure 4.56.

```
PublicClass Form21

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim a AsInteger
Dim b AsInteger

        a = TextBox1.Text
        b = TextBox2.Text

        TextBox3.Text = ((a ^ 1 / 2) * (b ^ 1 / 4)) / 3.7
EndSub

EndClass
```

Figure 4.56: The coding of runner size

After the code is generated, the runner size window is generated as show in Figure 4.57. It consists of empty columns where users will key in the data for the runner size calculation. When click answer button, calculation will refer to the program*TextBox3.Text = ((a ^ 1 / 2) * (b ^ 1 / 4)) / 3.7*. The result of the runner size is generated automatically.

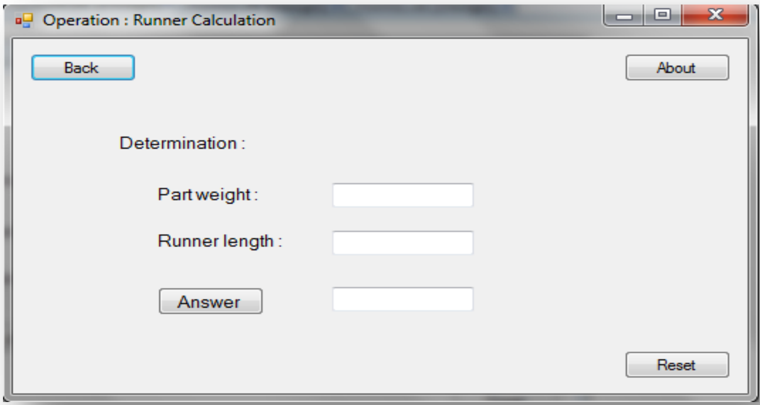


Figure 4.57: The runner size window

4.2.7.1 About window - The Runner Size

The About window is created to help the user to referring the definition and formula of the gate size by depth. The program is then generated as shown in Figure 4.58.

```
PublicClass Form21

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form22.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.58: The coding runner size

After the code is generated and the About window is created as show in Figure 4.59 by referring to the code "Form22.Show". From the form' properties, the definition and formula are written as shown in Figure 4.59.

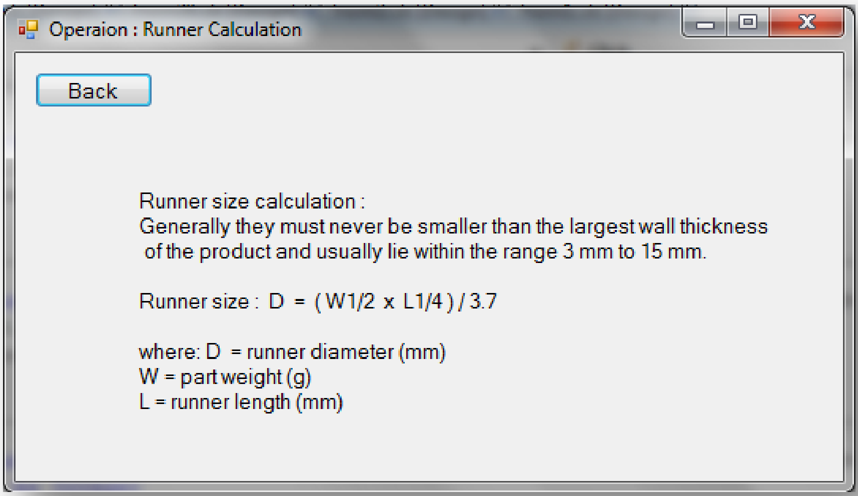


Figure 4.59: The About window

4.2.8 The Cycle Time Calculation - The Plunger and the Screw Type

For the cycle time calculation, the code for generate the window is input into the program. The program is then generated as shown in Figure 4.60.

```
PublicClass Form2

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form3.Show()
Me.Close()
EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Form5.Show()
Me.Close()
EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Form9.Show()
Me.Close()
EndSub

PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Form7.Show()
Me.Close()
EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Form16Show()
Me.Close()
EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
        Form21.Show()
Me.Close()
EndSub

PrivateSub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
        Form23.Show()
Me.Close()
EndSub

EndClass
```

Figure 4.60: The coding of cycle time

After the code is generated, the next window is created as shown in Figure 4.61. It shows the cycle time icon in the window when referring to the code "*Form23.show*".

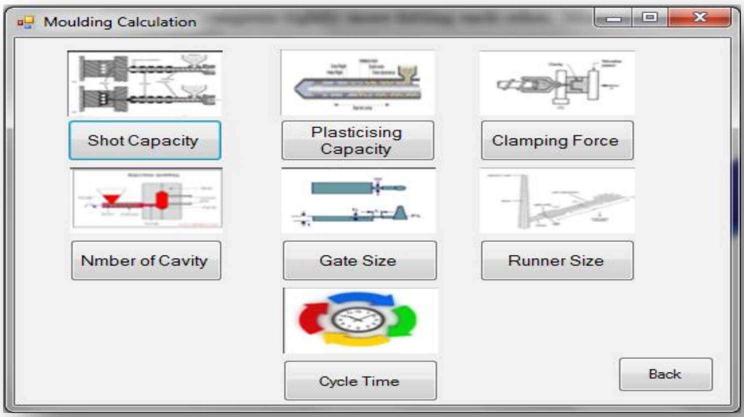


Figure 4.61: The cycle time window

To develop the cycle time calculation, the parameter is input into the program. The parameters are mass of shot and plasticizing capacity of the machine with the polymer being moulded, the program is then generated as shown in Figure 4.62.

```
PublicClass Form23

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger

        a = TextBox1.Text
        b = TextBox2.Text

        TextBox3.Text = (a * 3600) / b

EndSub

EndClass
```

Figure 4.62: The coding of the cycle time calculation

After the code is generated, the runner size window is generated as show in Figure 4.63. It consists of empty columns where users will key in the data for the runner size calculation. When click answer button, calculation will refer to the program:*TextBox3.Text = (a * 3600) / b*. The result of the runner size is generated automatically.

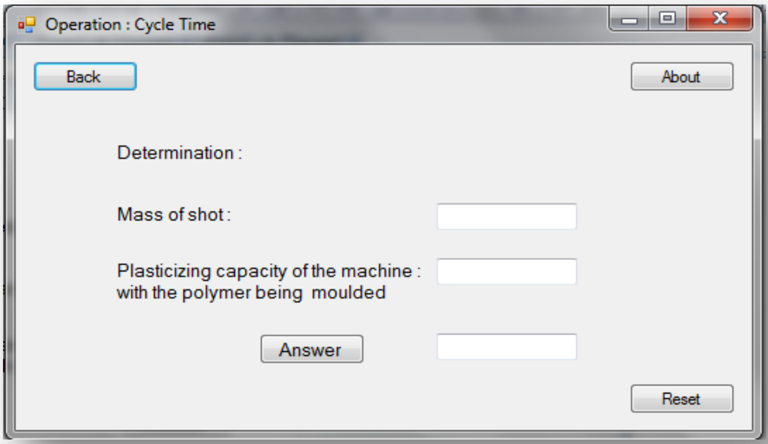


Figure 4.63: The window for cycle time calculation

4.2.8.1 About window - The Cycle Time

The About window is created to help the user to referring the definition and formula of the cycle time. The program is then generated as shown in Figure 4.64.

```
PublicClass Form23

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form24.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.64: The coding of cycle time

After the code is generated and the About window is created as show in Figure 4.65 by referring to the code "*Form24.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.65.

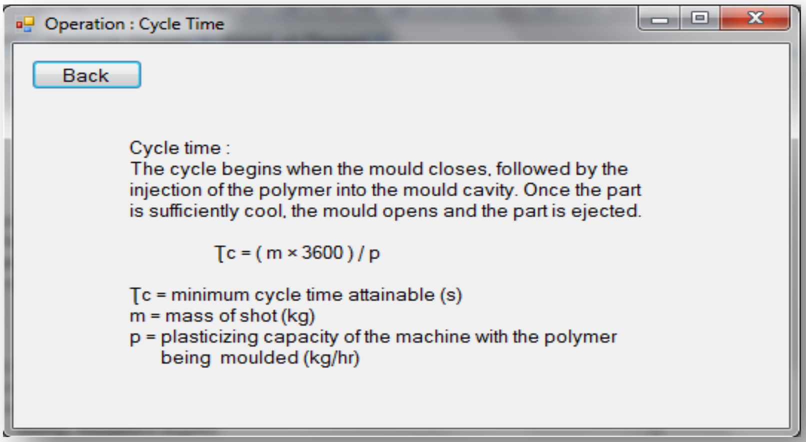


Figure 4.65: The About window

4.2.9 Front window - Screw Type

The front window is designed to have the features for selecting the type of the machine. One of the machine is screw type Hence, it is design to calculate shot capacity, plasticizing capacity, cycle time, clamping force, number of cavity, gate and runner size. For the screw type machine, in order to achieve the window design, the interface is design as shown in Figure 4.66 and consists of screw type button.

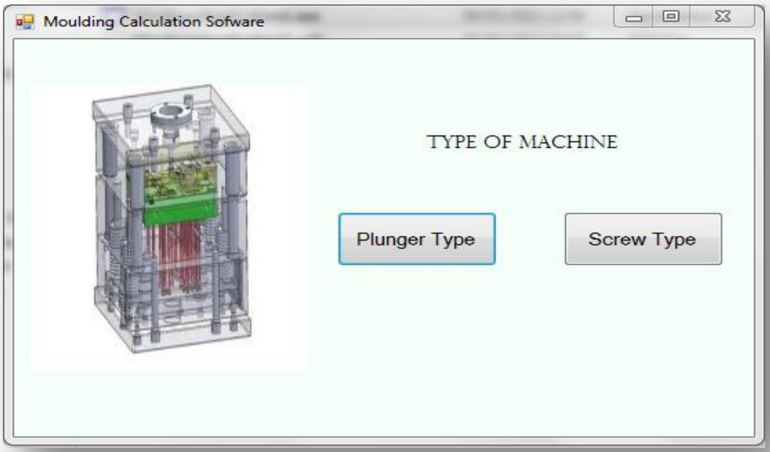


Figure 4.66: Screw type machine

4.2.10 Shot Capacity Calculation

The shot capacity calculation is input into the program. The program is then generated as shown in Figure 4.67.

```
PublicClass Form1

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        Form2.Show()
Me.Hide()

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        Form25.Show()
Me.Hide()

EndSub
```

Figure 4.67: The coding of the screw type window

After the code is generated, the next window is created as shown in Figure 4.68. It shows the shot capacity calculation icon in the window when referring to the code "*Form25.show*".

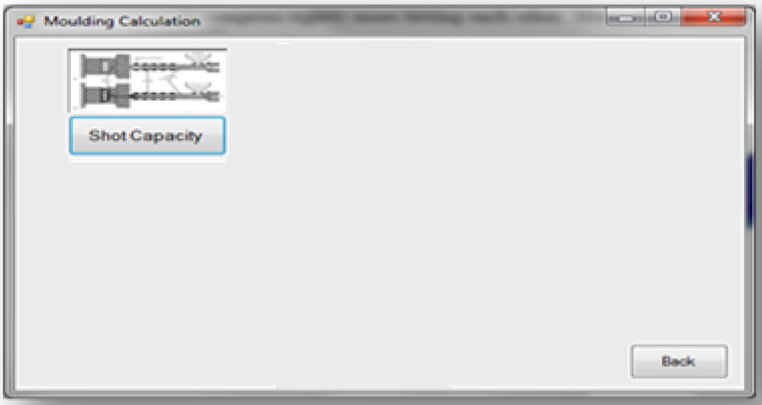


Figure 4.68: The process operation window

For the shot capacity calculation, the parameter is input into the program. The parameters are swept volume, density and constants. The program is then generated as show in Figure 4.69.

```
PublicClass Form25

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim a AsInteger
Dim b AsInteger
Dim c AsInteger

        a = TextBox1.Text
        b = TextBox2.Text
        c = TextBox3.Text

        TextBox4.Text = a * (b / c) * (d / f)

EndSub
EndClass
```

Figure 4.69: The code for shot capacity calculation

After the code is generated, the runner size window is generated as show in Figure 4.70. It consists of empty columns where users will key in the data for the runner size calculation. When click answer button, calculation will refer to the program:*TextBox4.Text = a * (b / c) * (d / f).* The result of the runner size is generated automatically

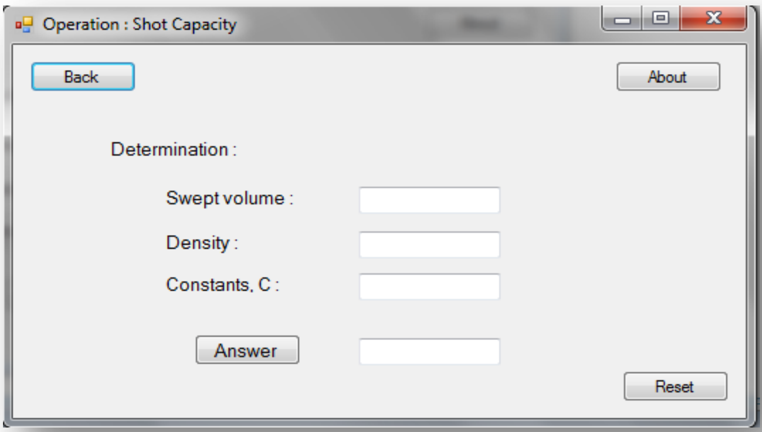


Figure 4.70: The shot capacity window

4.2.10.1 About window - The Shot Capacity

The About window is created to help the user to referring the definition and formula of the shot capacity. The program is then generated as shown in Figure 4.71.

```
PublicClass Form25

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Form26.Show()
Me.Close()

EndSub
EndClass
```

Figure 4.71: The coding cycle time

After the code is generated and the About window is created as show in Figure 4.64 by referring to the code "*Form26.Show*". From the form' properties, the definition and formula are written as shown in Figure 4.72.

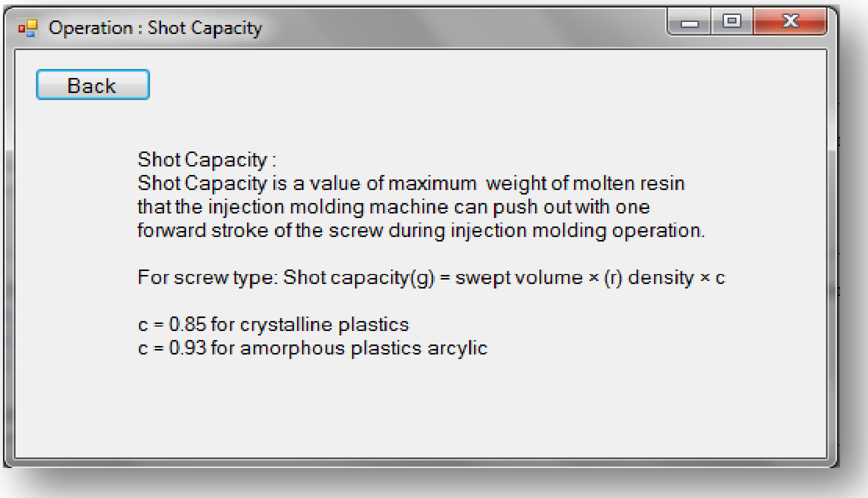


Figure 4.72: The About window

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The software of the mould calculation is developed during the project, with the aid of the Visual Basic 2008 programming. The coding is generated and the interface of the software is created. The formulae of each process in mould operation are used for the coding.

With the software, the calculation of the mould parameter is shortening where the users only need to key in the data to get the result or parameter. The time taken to calculate the process operation that involve in the project will be shorter than conventional calculation.

5.2 RECOMMENDATION

In the future, the database is needed for calculation software by using code block. This is because code block is simple programming and can install the equation easily.

The software integrated between the code block and visual basic program. This is because code block will act as its database and visual basic will be presented as graphic for this project.

This software should be integrated with drawing software such as Autodesk inventor. It should read the drawing and do the calculation automatically.

REFERENCES

1. ITERATIVE LEARNING CONTROL OF INJECTION MOLDING BY HEATHER L. HAVLICSEK, B.S., Case Western Reserve University, 1996
2. Injection Mold Design Guidelines, June1999, By Dr. Paul Engelmann and Bob Dealey for the Mold MarketingTask Group of the Copper,Development Association,
3. Engineered Materials Handbook Desk Edition, 1995, Michelle M. Gauthier, Editor, p 299-307.
4. Tool N Die for Marplex Polymers,C.G. Li, A.C.K. Mok, Department of Manufacturing Engineering,2009
5. Learn Visual Basic 6.0, © Lou Tylee, 1998 KIDware,15600 NE 8th, Suite B1-314,Bellevue, WA 98008,(206) 721-2556,FAX (425) 746-4655.
6. American National Standards Institute. American National Standard for Plastics Machinery –Horizontal Injection Molding Machines – Safety Requirements for Manufacture, Care, and Use,American National Standards Institute, 2007, 71 p. (ANSI/SPI B151.1).
7. Injection Molding, 2.810 Fall 2008,Professor Tim Gutowski.
8. Injection Molding-Process Description, Manufacturing Processes and Engineering,Prof. J.S. Colton © GIT 2009.
9. Automatic layout design of plastic injection mould cooling system, C.L. Li*, C.G. Li, A.C.K. Mok, Department of Manufacturing Engineering and Engineering Management, The City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China,Received 22 March 2004; received in revised form 29 July 2004; accepted 10 August 2004.
10. Press WH, Teukolsky SA, Vetterling W, Flannery BP. Numerical recipes in CCC. The art of scientific computing, 2nd ed. Cambridge: Cambridge University Press; 2002

Apendix 1

Apendix 2