Introduction

In computing, **memory** refers to a device that is used to store information for immediate use in a computer or related computer hardware device.[1] It typically refers to semiconductor memory, specifically metal–oxide–semiconductor (MOS) memory,[2][3] where data is stored within MOS memory cells on a silicon integrated circuit chip. The term "memory" is often synonymous with the term "primary storage". Computer memory operates at a high speed, for example random-access memory (RAM), as a distinction from storage that provides slow-to-access information but offers higher capacities. If needed, contents of the computer memory can be transferred to secondary storage; a very common way of doing this is through a memory management technique called "virtual memory". An archaic synonym for memory is **store**.[4]

The term "memory", meaning "primary storage" or "main memory", is often associated with addressable semiconductor memory, i.e. integrated circuits consisting of silicon-based MOS transistors[5], used for example as primary storage but also other purposes in computers and other digital electronic devices. There are two main kinds of semiconductor memory, volatile and non-volatile. Examples of non-volatile memory are flash memory (used as secondary memory) and ROM, PROM, EPROM and EEPROM memory (used for storing firmware such as BIOS). Examples of volatile memory are primary storage, which is typically dynamic random-access memory (DRAM), and fast CPU cache memory, which is typically static random-access memory (SRAM) that is fast but energy-consuming, offering lower memory areal density than DRAM.

History and Evolution

Computer is an appliance which can manipulate data in accordance with a list of instructions. It is a type of data processing system.

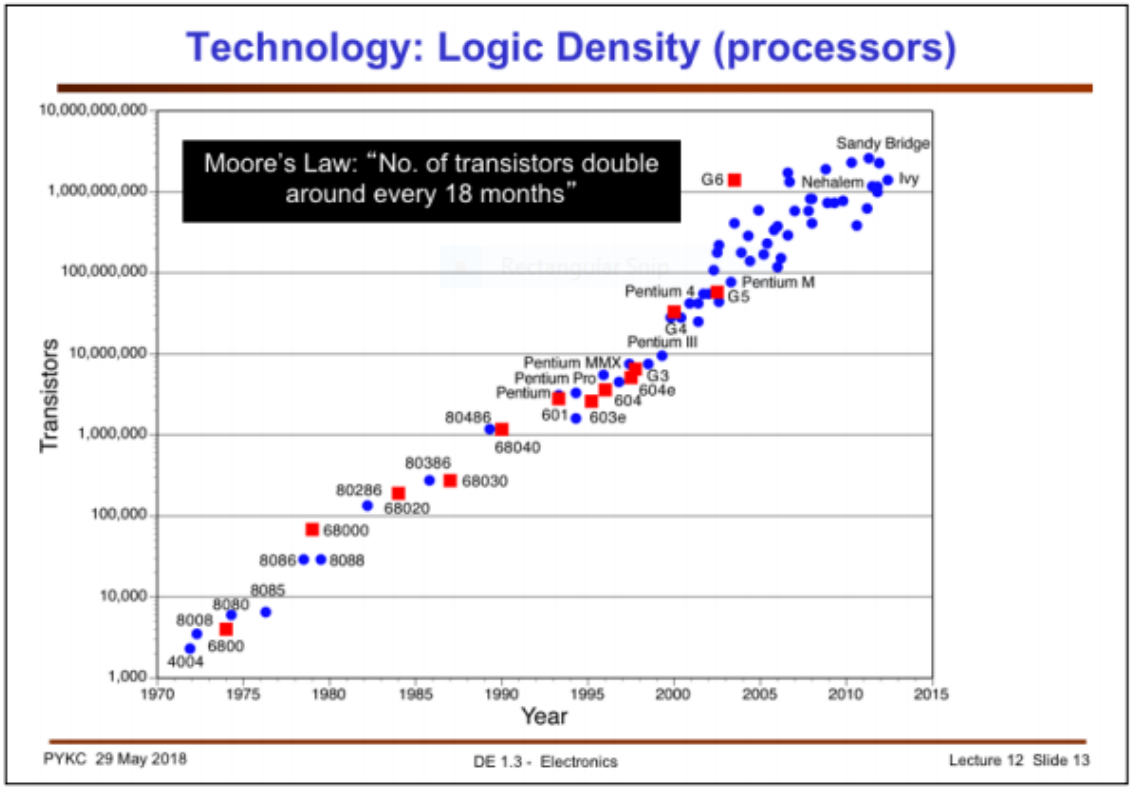
Today the computer has become indispensable in every household and comes in numerous physical forms. The first modern day computer was developed in the mid-20th century, though the concept of computer and various similar machines existed before. It actually started in 1837, when `The Analytical Engine`, the first fully programmable mechanical computer, was designed by Charles Babbage.

The Earlier versions were huge and bulky and used up a whole room. The device EDSAC (Electronic Delay Storage Automatic Calculator) was one of the very first computers that could implement the stored program architecture. In comparison to them, modern computers have tiny integrated circuits and are much better in terms of capacity and speed as well as accuracy.

What makes computers highly versatile and distinguishable from other appliances is that it can be programmed. A calculator can only calculate, just like a washing machine can only wash, but a computer can be programmed to do any kind of job. Software programs are a list of instructions that can be stored and executed by the computer.

A general computer has four major sections, the arithmetic and logical unit or ALU, the control unit or CU, memory and the interface for input and output devices. These parts of computer are interconnected by busses. The ALU, control unit, registers, and the interface for input and output devices are collectively known as the central processing unit or CPU. The Early CPUs used to be composed of different separate components, however since the 1975; the CPUs are being constructed on a single integrated circuit, the microprocessor.

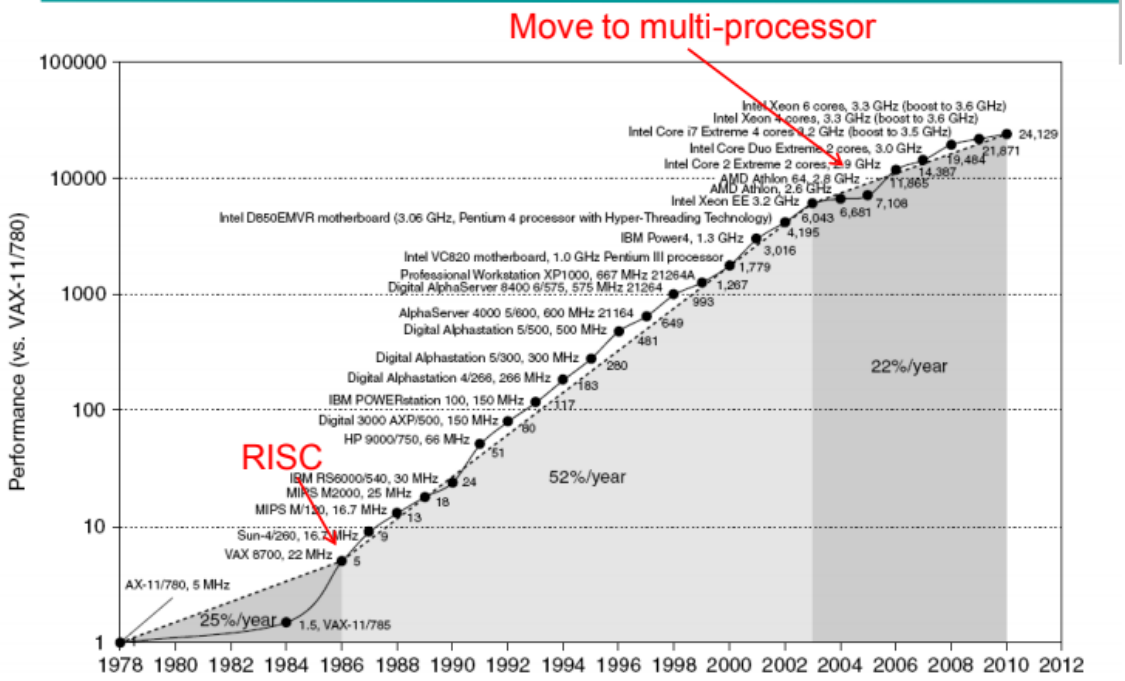
The earlier version of computer used Magnetic core memory but today it has been replaced by the semiconductor memory. The main memory of Computer is divided in two parts, RAM or random access memory and ROM or read-only memory. RAM can be written and read anytime through CPU commands. On the other hand, ROM is pre-loaded with software and data that never changes, and it can be only read by the CPU.

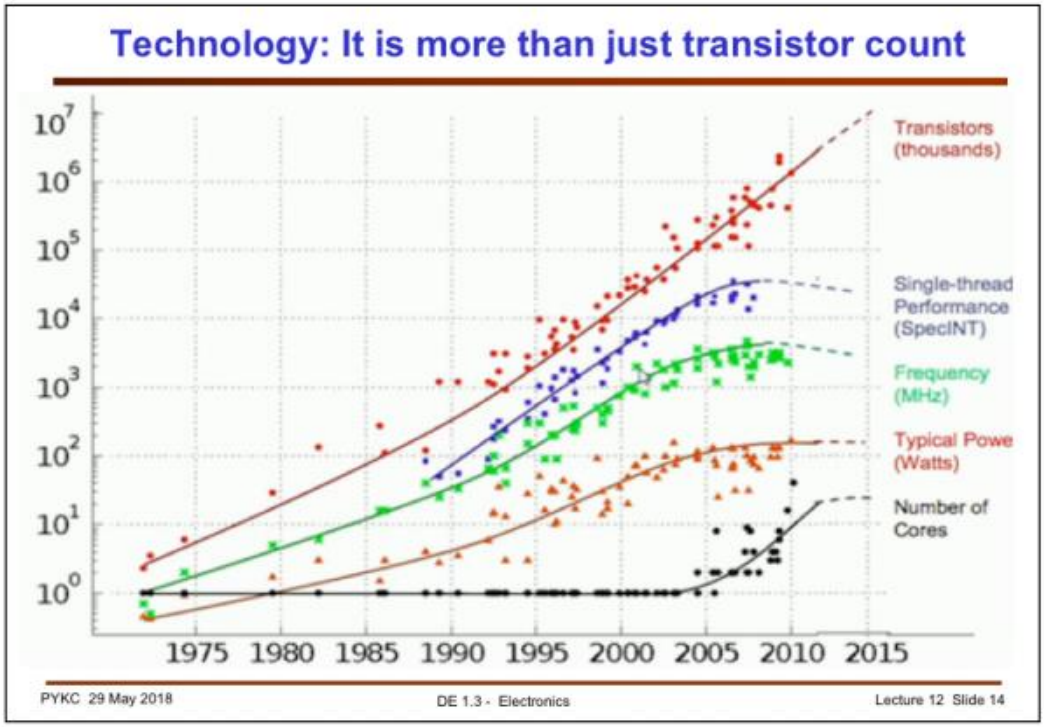


Gordon Moore from Intel famously observed that transistor density on silicon doubles every year to year and a half. That became the Moore's law. It drove the semiconductor industry for the past four decades, and it became a self-fulfilling prophesy.

Here is a lot of how transistor density (i.e. no of transistors on a processor chip) over the years. Note that the y-axis is in log scale.

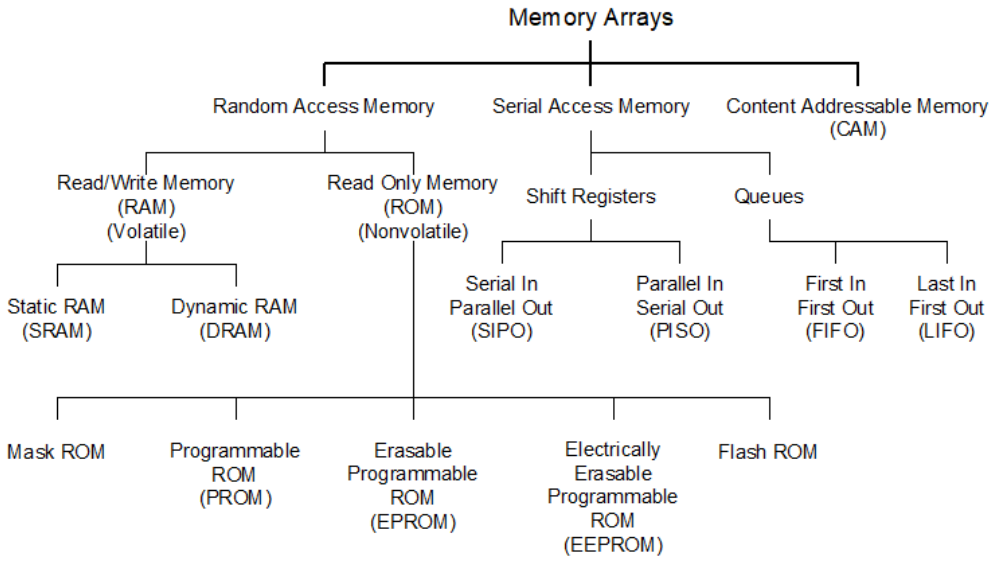
Single Processor Performance





We often heard people say that this is the end of Moore's law. I think this view is incomplete. If fact if you look at the plot here, transistor density is still going up as Moore has predicted. However what becomes much more significant is NOT transistor density, but other factors including power consumption (by each chip) and the performance you get from processor. These other factors are NOT increasing. They are leveling out!

Memory



Memory are basically of 2 types

1. **Votalile Memory**: It's retain the information as long as power is given to it. Power cut off lead to loss of data.
2. **Non volatile Memory**: It's stores the data even power is off.

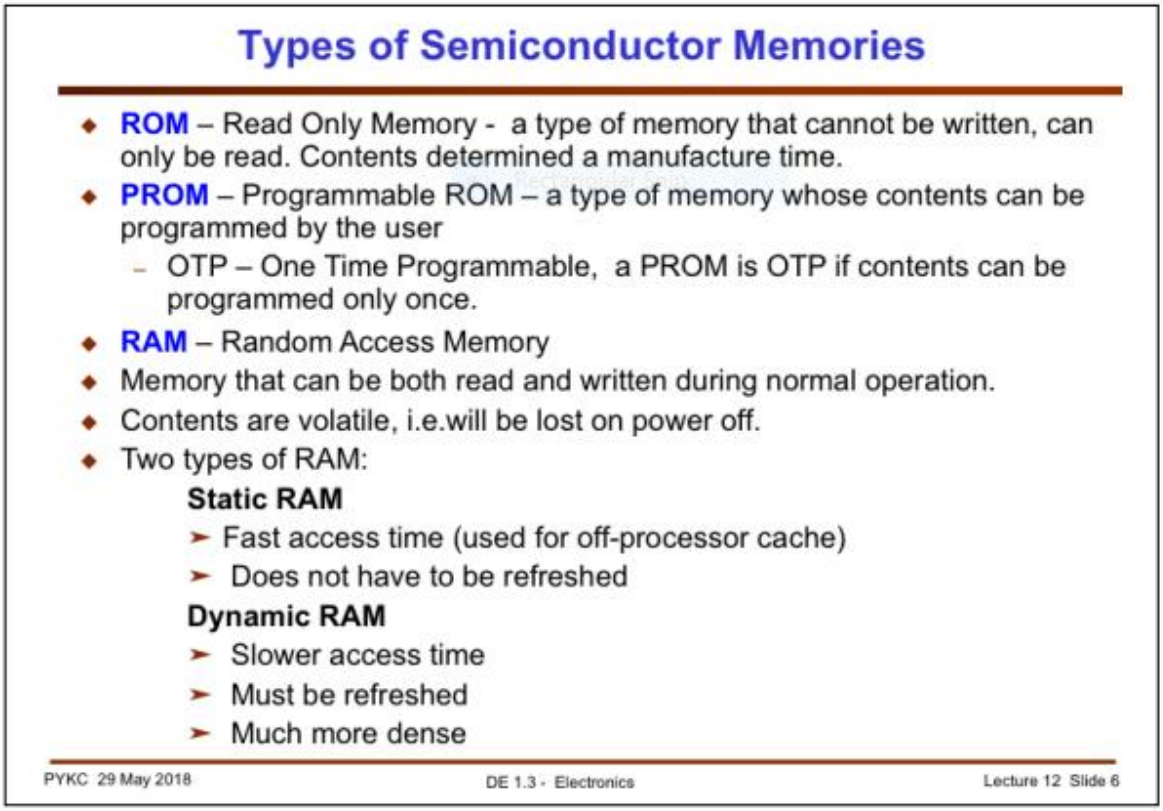
History

Efforts began in the late 1940s to find non-volatile memory. Magnetic-core memory allowed for recall of memory after power loss. It was developed by Frederick W. Viehe and An Wang in the late 1940s, and improved by Jay Forrester and Jan A. Rajchman in the early 1950s, before being commercialised with the Whirlwind computer in 1953.[6] Magnetic-core memory would become the dominant form of memory until the development of MOS semiconductor memory in the 1960s[7].

Semiconductor memory began in the early 1960s with bipolar memory, which used bipolar transistors.[7] Bipolar semiconductor memory made from discrete devices was first shipped by Texas Instruments to the United States Air Force in 1961. The same year, the concept of solid-state memory on an integrated circuit (IC) chip was proposed by applications engineer Bob Norman at Fairchild Semiconductor.[8] The first bipolar semiconductor memory IC chip was the SP95 introduced by IBM in 1965.[7] While bipolar memory offered improved performance over magnetic-core memory, it could not compete with the lower price of magnetic-core, which remained dominant up until the late 1960s.[7] Bipolar memory failed to replace magnetic-core memory because bipolar flip-flop circuits were too large and expensive.[9]\

MOS Memory

The invention of the MOSFET (metal–oxide–semiconductor field-effect transistor, or MOS transistor), by Mohamed M. Atalla and Dawon Kahng at Bell Labs in 1959,[5] enabled the practical use of metal–oxide–semiconductor (MOS) transistors as memory cell storage elements, a function previously served by magnetic cores.[10] MOS memory was developed by John Schmidt at Fairchild Semiconductor in 1964.[11][12] In addition to higher performance, MOS semiconductor memory was cheaper and consumed less power than magnetic core memory.[11] In 1965, J. Wood and R. Ball of the Royal Radar Establishment proposed digital storage systems that use CMOS (complementary MOS) memory cells, in addition to MOSFET power devices for the power supply, switched cross-coupling, switches and delay line storage.[13] The development of silicon-gate MOS integrated circuit (MOS IC) technology by Federico Faggin at Fairchild in 1968 enabled the production of MOS memory chips.[14] NMOS memory was commercialized by IBM in the early 1970s.[15] MOS memory overtook magnetic core memory as the dominant memory technology in the early 1970s.[11]

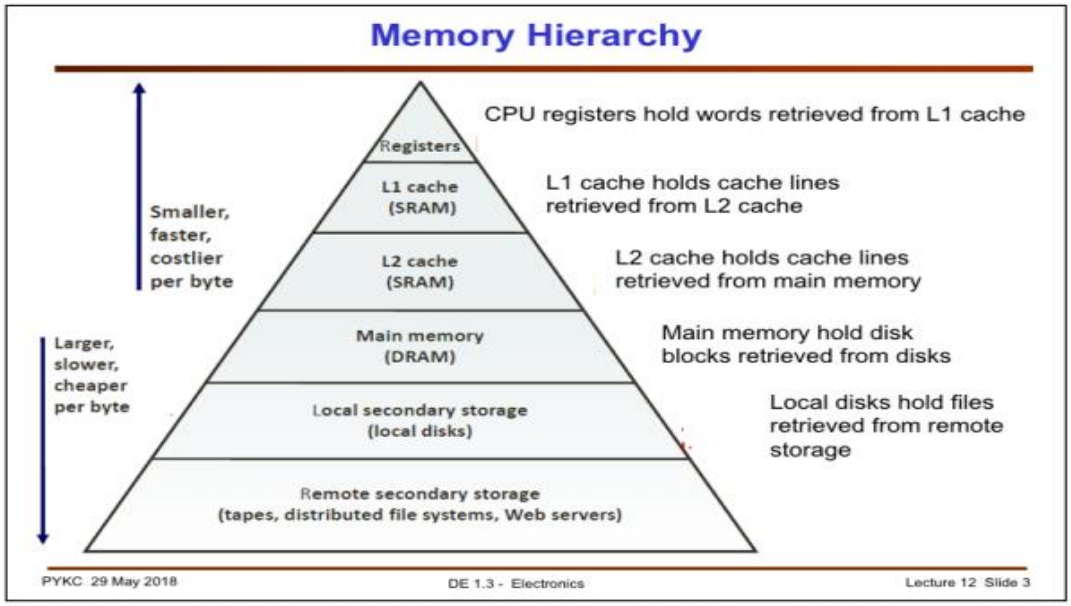


Volatile Memory

1. Volatile memory is computer memory that requires power to maintain the stored information. Most modern semiconductor volatile memory is either static RAM (SRAM) or dynamic RAM (DRAM). SRAM retains its contents as long as the power is connected and is simpler for interfacing, but uses six transistors per bit. Dynamic RAM is more complicated for interfacing and control, needing regular refresh cycles to prevent losing its contents, but uses only one transistor and one capacitor per bit, allowing it to reach much higher densities and much cheaper per-bit costs.[1][22][36]
2. SRAM is not worthwhile for desktop system memory, where DRAM dominates, but is used for their cache memories. SRAM is commonplace in small embedded systems, which might only need tens of kilobytes or less. Forthcoming volatile memory technologies that aim at replacing or competing with SRAM and DRAM include Z-RAM and A-RAM.

Non Volatile Memory

1. Non-volatile memory is computer memory that can retain the stored information even when not powered. Examples of non-volatile memory include read-only memory (see ROM), flash memory, most types of magnetic computer storage devices (e.g. hard disk drives, floppy disks and magnetic tape), optical discs, and early computer storage methods such as paper tape and punched cards.[36]
2. Forthcoming non-volatile memory technologies include FERAM, CBRAM, PRAM, STT-RAM, SONOS, RRAM, racetrack memory, NRAM, 3D XPoint, and millipede memory.



Radom Access Memory

Random access memory (also hyphenated as *random-access memory*), usually known by its acronym **RAM**, is a class of media used in computers for data storage and retrieval. A RAM device is designed to allow data to be read or written in any order—that is, "at random." In addition, the speed at which a set of data can be accessed is independent of its location in the device.

In today's computers, the main memory takes the form of a RAM device, which is usually an integrated circuit containing millions of "memory cells." This memory stores software programs as well as other data that are actively being used. RAM is a volatile type of memory, where the information is lost when the power is switched off.

RAM stands in contrast to sequential access memory (SAM), where the reading or writing functions are carried out sequentially. For example, a magnetic tape stores data that can be accessed in only a sequential manner.

Types of RAM

Various types of RAM have been developed. The most common type of semiconductor RAM is called "dynamic random access memory," or DRAM. In this case, each memory cell is formed by pairing a transistor with a capacitor, and each bit of data is stored as an electric charge in a capacitor.[1] There are millions of memory cells in each memory chip. In a DRAM chip, the content (or charge) at every location is held for a fraction of a second and needs to be refreshed repeatedly.[2]

A second type of semiconductor RAM is called "static random access memory," or SRAM. Here, a bit of data is stored in the state of an electronic "flip-flop." An SRAM chip retains its contents at all locations as long as the power supply is present.[2]

Many other types of memory can be classified as RAM as well, including most types of read only memory (ROM) and a kind of flash memory called *NOR-Flash*. RAM of the *read-only* type uses a metal mask to permanently enable/disable selected transistors, instead of storing a charge in them. Some types of RAM have circuitry to detect and/or correct random faults called *memory errors* in the stored data.

Dynamic Random Access Memory

Dynamic random access memory (DRAM) is a type of semiconductor memory that is typically used for the data or program code needed by a computer processor to function. DRAM is a common type of random access memory (RAM) that is used in personal computers (PCs), workstations and servers. Random access allows the PC processor to access any part of the memory directly rather than having to proceed sequentially from a starting place. RAM is located close to a computer's processor and enables faster access to data than storage media such as hard disk drives and solid-state drives.

History

One of the first uses of DRAM was in a Toshiba calculator in 1965 -- using a capacitive form of DRAM that was made from bipolar memory cells. That same year, IBM created a 16 bit silicon memory chip. However, at this time, the bipolar DRAM that was in use could not compete against magnetic-core memory. This stayed true of DRAM until the invention if the metal-oxide-semiconductor field-effect transistor (MOSFET), which lead to the metal-oxide-semiconductor DRAM -- or MOS DRAM. The patent for MOS DRAM was granted in 1968. 1969 saw Intel develop DRAM that used a three transistor cell.

How does DRAM work?

Memory is made of bits of data or program code that are arranged in a two-dimensional grid. DRAM will store bits of data in what's called a storage, or

memory cell, consisting of a capacitor and a transistor. The storage cells are typically organized in a rectangular configuration. When a charge is sent through a column, the transistor at the column is activated. A DRAM storage cell is dynamic, meaning that it needs to be refreshed or given a new electronic charge every few milliseconds to compensate for charge leaks from the capacitor.

Types of DRAM

There are many types of DRAM that can be used in a device. Some examples include the following:

- Synchronous DRAM (SDRAM) syncs memory speeds with CPU clock speeds, letting the memory controller know the CPU clock cycle. This allows the CPU to perform more instructions at a time.
- Rambus DRAM (RDRAM) was more widely used in the early 2000s for graphics cards.
- Double Data Rate SDRAM (DDR SDRAM) almost doubles the bandwidth in data rate of SDRAM by using double pinning. This process allows for data to transfer on rising and falling edges of a clock signal. It has been available in different iterations over time, including DDR2 SDRAM, DDR3 SDRAM and DDR4 SDRAM.
- Fast page mode DRAM (FPM DRAM) gives higher performance than other DRAM types through focusing on fast page access.
- Extended data out DRAM (EDO DRAM) improves the time to read from memory on microprocessors, such as the Intel Pentium.

Major DRAM manufacturers include Samsung, Rambus, PNY Technologies and SK Hynix.

Types of DRAM packages

There are two main types of DRAM packaging: single inline memory module (SIMM) and dual inline memory module (DIMM). Single inline memory module packaging is considered obsolete now and was used in the 1980s to 1990s. SIMMs came in 30 and 72 pin sets and typically had 32 bit data transfer rates. DIMMs, on the other hand, are commonly used now and are dual inline -- meaning that they have pins on both sides of the chip. DIMMS commonly have 168 pin connectors -- or more -- and support a 64 bit data transfer rate.

DRAM package types for DIMMs are set as different integrated circuit architectures. Some of these include the following:

- Unbuffered DIMMs (UDIMMs) are commonly used on desktops and laptops. These cost less and run faster, but are less stable.
- Registered DIMMs (RDIMMs) are commonly used with servers. These are more stable and reduce strain on a CPUs memory controller.
- Fully buffered DIMMs (FB-DIMMs) are used in larger memory systems. These are more reliable since they can improve error detection methods and maintain signal integrity.

Advantages

The main advantages of DRAM include the following:

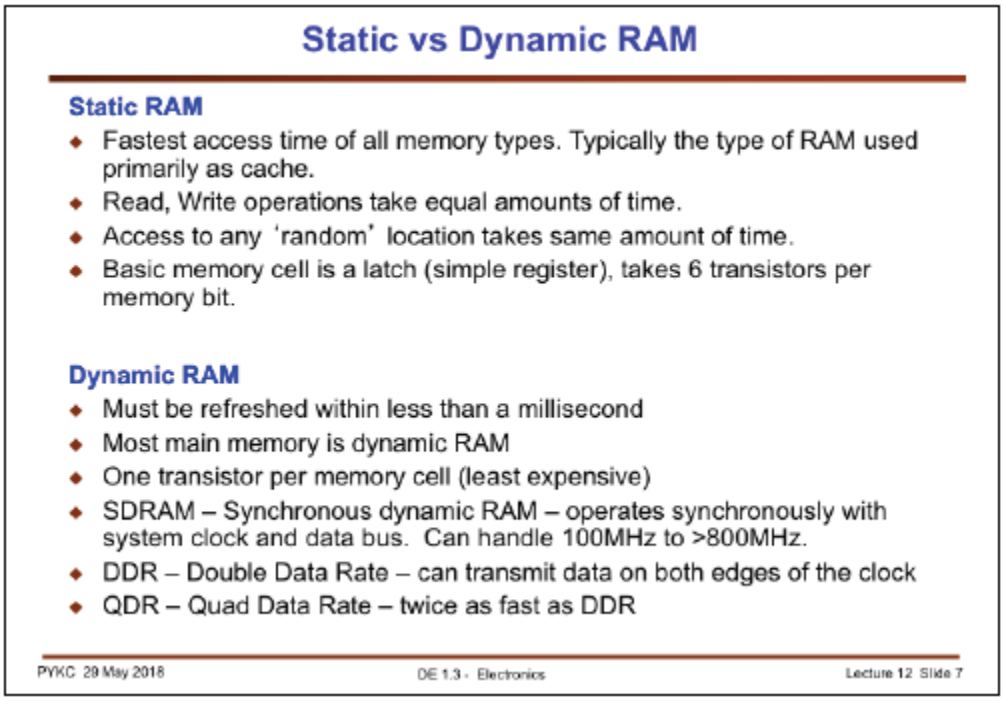
- Its design is simple, only requiring one transistor.
- The cost is low in comparison to alternative types of memory such as SRAM.
- It provides higher density levels.
- More data can be stored using DRAM.

- Memory can be refreshed and deleted while a program is running.

Disadvantages

The main disadvantages of DRAM include the following:

- Memory is volatile.
- Power consumption is high relative to other options.
- Manufacturing is complex.
- Data in storage cells needs to be refreshed.
- It is slower than SRAM.



Static Random Access Memory

SRAM or Static Random Access Memory is a form of semiconductor memory widely used in electronics, microprocessor and general computing applications. This form of semiconductor memory gains its name from the fact that data is held in there in a static fashion, and does not need to be dynamically updated as in the case of DRAM memory. While the data in the SRAM memory does not need to be refreshed dynamically, it is still volatile, meaning that when the power is removed from the memory device, the data is not held, and will disappear.

A static random access memory (SRAM) contains *N* registers addressed by log *N* address bits *A*. SRAM is so named because the underlying flip-flops refresh themselves and so are "static." Besides flip-flops, an SRAM also needs a decoder that decodes A into a unary value used to select the right register. Accessing an SRAM on-chip is only slightly slower than accessing a register, because of the added decode delay. At the time of writing, it was possible to obtain on-chip SRAMs with 0.5-nsec access times. Access times of of 1–2 nsec for on-chip SRAM and 5–10 nsec for off-chip SRAM are common.

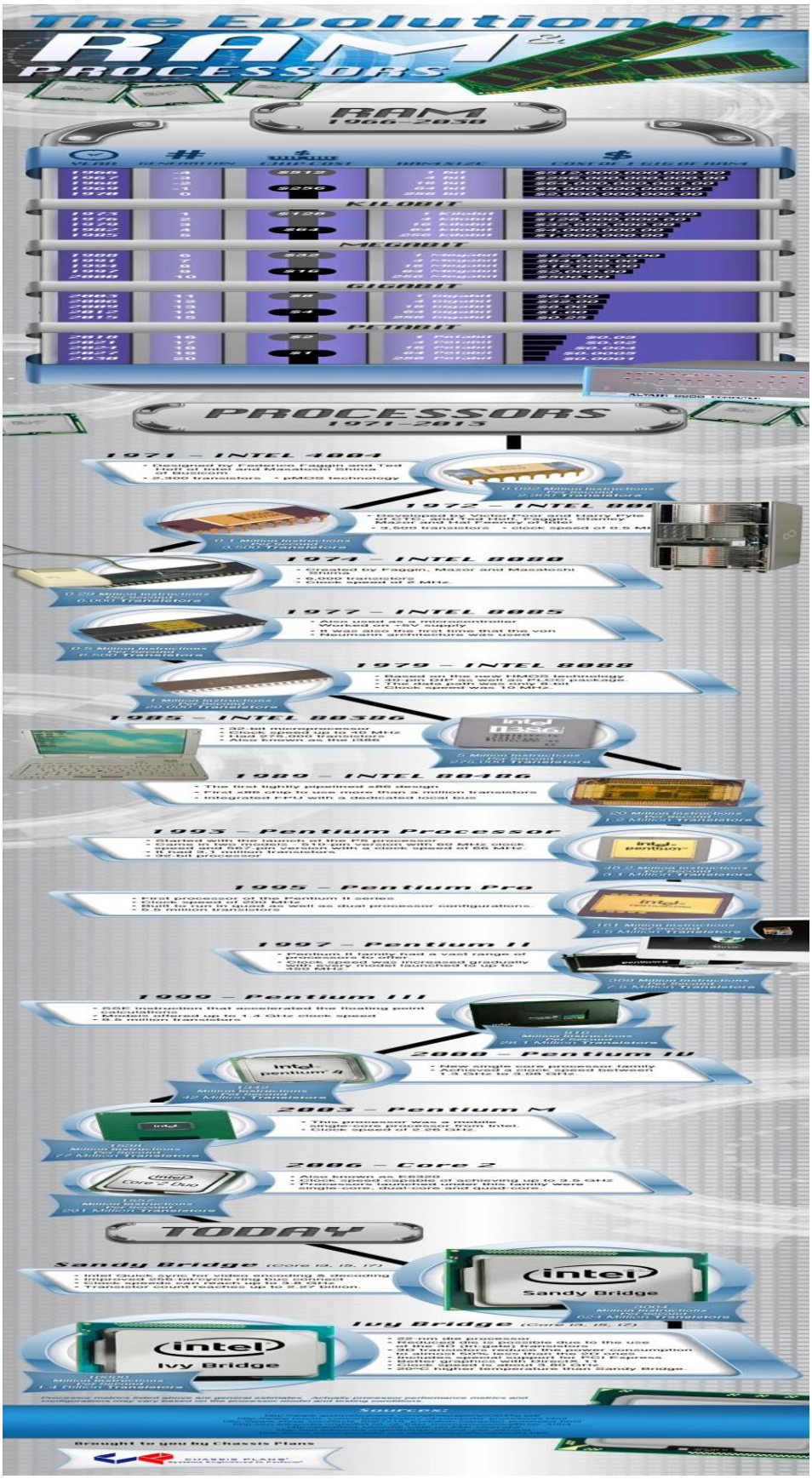
SRAM basics

There are two key features to SRAM - Static random Access Memory, and these set it out against other types of memory that are available:

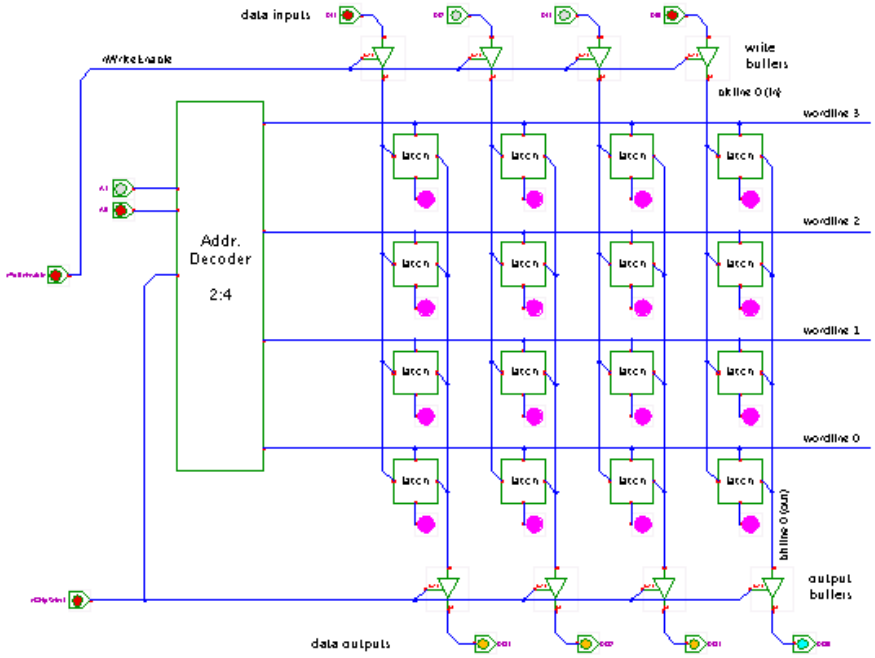
- ***The data is held statically:*** This means that the data is held in the semiconductor memory without the need to be refreshed as long as the power is applied to the memory.
- ***SRAM memory is a form of random access memory:*** A random access memory is one in which the locations in the semiconductor memory can be written to or read from in any order, regardless of the last memory location that was accessed.

The circuit for an individual SRAM memory cell comprises typically four transistors configured as two cross coupled inverters. In this format the circuit has two stable states, and these equate to the logical "0" and "1" states. In addition to the four transistors in the basic memory cell, and additional two transistors are required to control the access to the memory cell during the read and write operations. This makes a total of six transistors, making what is termed a 6T memory cell. Sometimes further transistors are used to give either 8T or 10T memory cells. These additional transistors are used for functions such as implementing additional ports in a register file, etc for the SRAM memory.

Although any three terminal switch device can be used in an SRAM, MOSFETs and in particular CMOS technology is normally used to ensure that very low levels of power consumption are achieved. With semiconductor memories extending to very large dimensions, each cell must achieve a very low levels of power consumption to ensure that the overall chip does not dissipate too much power.


The Evolution of
RAM
PROCESSORS
RAM
1966 - 2030
KILOBIT
MEGABIT
GIGABIT
TERABIT
PROCESSORS
1971 - 2013
1971 – INTEL 4004
1972 – INTEL 800
1974 – INTEL 8080
1977 – INTEL 8085
1979 – INTEL 8088
1985 – INTEL 80386
1989 – INTEL 80486
1993 – Pentium Processor
1995 – Pentium Pro
1997 – Pentium II
1999 – Pentium III
2000 – Pentium 4
2003 – Pentium M
2006 – Core 2
TODAY
Sandy Bridge (Core i3, i5, i7)
Ivy Bridge (Core i3, i5, i7)
Sources:
Brought to you by Chassis Plans

RAM (random access memory) structure



Read Only Memory(ROM)

Read-only memory (**ROM**) is a type of non-volatile memory used in computers and other electronic devices. Data stored in ROM cannot be electronically modified after the manufacture of the memory device. Read-only memory is useful for storing software that is rarely changed during the life of the system, also known as firmware. Software applications (like video games) for programmable devices can be distributed as plug-in cartridges containing read-only memory.

Strictly, *read-only memory* refers to memory that is hard-wired, such as diode matrix or a mask ROM integrated circuit, which cannot be electronically[a] changed after manufacture. Although discrete circuits can be altered in principle, through the addition of bodge wires and/or the removal or replacement of components, integrated circuits (ICs) cannot.

Erasable programmable read-only memory (EPROM) and electrically erasable programmable read-only memory (EEPROM) can be erased and re-programmed, but usually this can only be done at relatively slow speeds, may require special equipment to achieve, and is typically only possible a certain number of times.[1]

Types of ROM

Semiconductor-based

Classic *mask-programmed ROM* chips are integrated circuits that physically encode the data to be stored, and thus it is impossible to change their contents after fabrication. Other types of non-volatile solid-state memory permit some degree of modification:

- *Programmable read-only memory* (PROM), or *one-time programmable ROM* (OTP), can be written to or *programmed* via a special device called a *PROM programmer*. Typically, this device uses high voltages to permanently destroy or create internal links (fuses or antifuses) within the chip. Consequently, a PROM can only be programmed once.
- *Erasable programmable read-only memory* (EPROM) can be erased by exposure to strong ultraviolet light (typically for 10 minutes or longer), then rewritten with a process that again needs higher than usual voltage applied. Repeated exposure to UV light will eventually wear out an EPROM, but the *endurance* of most EPROM chips exceeds 1000 cycles of erasing and reprogramming. EPROM chip packages can often be identified by the prominent quartz "window" which allows UV light to enter. After programming, the window is typically covered with a label to prevent accidental erasure. Some EPROM chips are factory-erased before they are packaged, and include no window; these are effectively PROM.
- *Electrically erasable programmable read-only memory* (EEPROM) is based on a similar semiconductor structure to EPROM, but allows its entire contents (or selected *banks*) to be electrically erased, then rewritten electrically, so that they need not be removed from the computer (whether general-purpose or an embedded computer in a camera, MP3 player, etc.). Writing or *flashing* an EEPROM is much slower (milliseconds per bit) than reading from a ROM or writing to a RAM (nanoseconds in both cases).
  - *Electrically alterable read-only memory* (EAROM) is a type of EEPROM that can be modified one bit at a time. Writing is a very slow process and again needs higher voltage (usually around 12 V) than is used for read access. EAROMs are intended for applications that require infrequent and only partial rewriting. EAROM may be used as non-volatile storage for critical system setup information; in many applications, EAROM has been supplanted by CMOS RAM supplied by mains power and backed-up with a lithium battery.
  - *Flash memory* (or simply *flash*) is a modern type of EEPROM invented in 1984. Flash memory can be erased and rewritten faster than ordinary EEPROM, and newer designs feature very high endurance (exceeding 1,000,000 cycles). Modern NAND flash makes efficient use of silicon chip area, resulting in individual ICs with a capacity as high as 32 GB as of 2007; this feature, along with its endurance and physical durability, has allowed NAND flash to replace magnetic in some applications (such as USB flash drives). Flash memory is sometimes called *flash ROM* or *flash EEPROM* when used as a replacement for older ROM types, but not in applications that take advantage of its ability to be modified quickly and frequently.

By applying write protection, some types of reprogrammable ROMs may temporarily become read-only memory.

Programmable ROM

PROM or **programmable ROM** (**programmable read-only memory**) is a computer memory chip that can be programmed once after it is created. Once the PROM is programmed, the information written is permanent and cannot be erased or deleted. PROM was first developed by Wen Tsing Chow in 1956. An example of a PROM is a computer BIOS in early computers. Today, PROM in computers has been replaced by EEPROM.

When the PROM is created, all bits read as "1." During the programming, any bit needing to be changed to a "0" is etched or burned into the chip using a **gang programmer**. Below is an example of a gang programmer from Advin that programs multiple ROM chips at one time.

If a PROM is programmed with an error or needs updated, the chip is discarded and a new PROM is created, replacing the old chip. A variation of the PROM is an EPROM, which is a PROM that can be erased and reprogrammed without being replaced.

Short for **Erasable Programmable Read-Only Memory**, **EPROM** is a non-volatile memory chip that was invented by Dov Frohman in 1971 while at Intel that can only be read. If exposed to ultraviolet light, an EPROM can be reprogrammed if needed, but otherwise does not accept or save any new data. Hardware manufactures use EPROM when it may be needed that the data contained on the EPROM needs to be changed. An EPROM chip is distinguishable by a small quartz crystal (not glass) circle window that exposes the chip so that can be reprogrammed. The picture on this page is an example of an Intel 8048 made by NEC and is an example of an EPROM chip.

Electrically erasable programmable read-only memory (EEPROM)

Short for **electrically erasable programmable read-only memory**, **EEPROM** is a PROM that can be erased and reprogrammed using an electrical charge. EEPROM was developed by George Perlegos while at Intel in 1978 and unlike most memory inside a computer, it remembers its data without power.

EEPROM was a replacement for PROM and EPROM chips and is used for later computer's BIOS that were built after 1994. Having a computer with an EEPROM allows the user to update the computer BIOS without having to open the computer or remove any chips.

Hard Drive

What is Hard Drive

Your hard drive, also sometimes called your hard disk drive, HD, or HDD is the permanent storage device on your computer. It's non-volatile, which means that it will store the information regardless of whether it's turned on or off. Things like your system settings or time zone are housed here on most computers.

Your hard drive is made up of one or more platters where a magnetic head writes data, and it lives inside an air-sealed casing that can be inside or outside of your computer.

If the casing is stored inside, these are called internal hard disks, and they live inside the drive bay and connect directly to your motherboard using a cable. Outside, or external hard disks plug into your computer through a USB port and are another way to store data permanently.

Every computer has a hard drive, and while you can upgrade or swap the standard model from yours, it will always be used to store the files that manage your operating system, critical software programs, and any personal data you save.

Content-addressable memory (CAM)

Content-addressable memory (CAM) is computer memory that operates like a hardware search engine for search-intensive applications.

History-

CAM is the hardware embodiment of what in software terms would be called an associative array. The data word recognition unit was proposed by Dudley Allen Buck in 1955.A major interface definition for CAMs and other network search engines (NSEs) was specified in an interoperability agreement called the Look-Aside Interface (LA-1 and LA-1B) developed by the Network Processing Forum, which later merged with the Optical Internetworking Forum (OIF). Numerous devices have been produced by Integrated Device Technology, Cypress Semiconductor, IBM, Broadcom and others to the LA interface agreement. On December 11, 2007, the OIF published the serial lookaside (SLA) interface agreement.

Working Principle-

CAM is capable of searching its entire contents in a single clock cycle. It does that by pairing the SRAM-based memory with additional logic comparison circuitry that is active on every clock

cycle.The way CAM functions is almost the opposite of random access memory (RAM). To retrieve data residing on RAM, the OS provides the memory address where the data is stored. Data stored on CAM, on the other hand, can be accessed by

searching for the content itself, and the memory retrieves the addresses where that content can be found. Because of its parallel nature, CAM is much faster than RAM for searching.

Ternary CAM (TCAM) adds a third state to RAM, beyond binary, for a wildcard functionality that provides for variable characters in searches and adds additional complexity to the circuits.Because of their low capacities, high power usage and consequent heat dissipation requirements, CAM and TCAM remain isolated to specialized applications such as Internet routers and switches, where they increase the speed of route look-up, packet classification, packet forwarding and access control list-based commands.

Applications-CAM is much faster than RAM in virtually all search applications. ... Consequently, CAM is used only in specialized applications where searching speed cannot be accomplished using a less costly method.

2. Problem :- Implementation of KMP matching algorithm with FSM. Here only 4 character are used in a string for implementation (Overlapping allowed in pattern matching)

Encoding

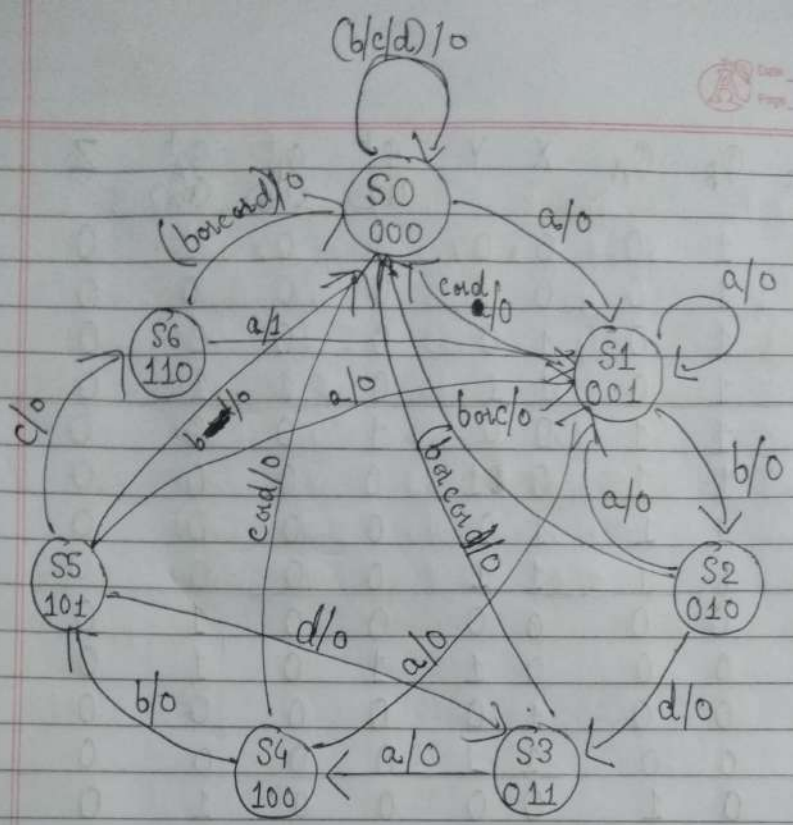
Alphabet	X	Y
a	0	0
b	0	1
c	1	0
d	1	1

Pattern to be search :- abdabca

Output Z = 1 when search is found.

Defining different States

State	Matched till
S0 (000)	No Match Yet
S1 (001)	a
S2 (010)	ab
S3 (011)	abd
S4 (100)	abda
S5 (101)	abdab
S6 (110)	abdabc



State Diagram

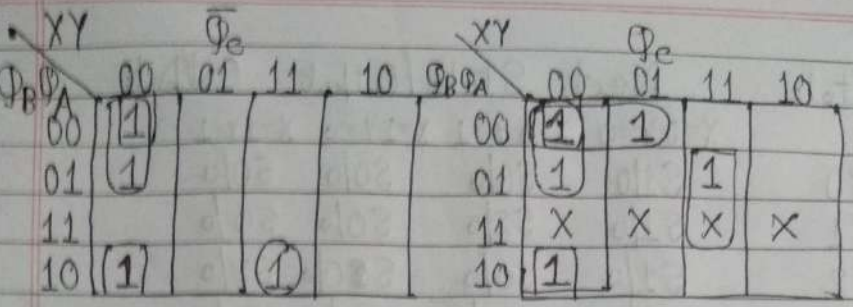
Truth Table

Q_C	Q_B	Q_A	X	Y	Q_C^+ (D_C)	Q_B^+ (D_B)	Q_A^+ (D_A)	Z
0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0

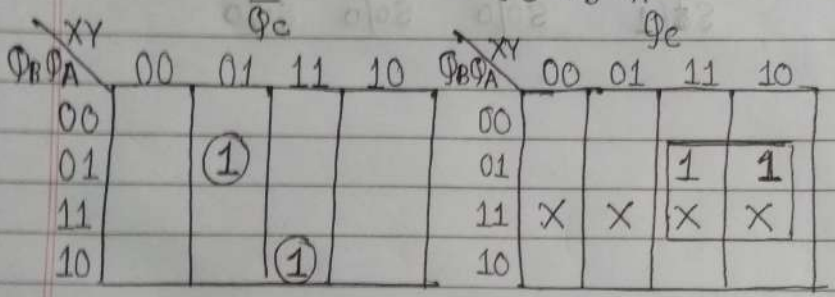
Q_C	Q_B	Q_A	X	Y	Q_C^+ (D_C)	Q_B^+ (D_B)	Q_A^+ (D_A)	Z
0	1	0	0	0	0	0	1	0
0	1	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0	0
0	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	1	0
1	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0
1	0	1	0	0	0	0	1	0
1	0	1	0	1	0	0	0	0
1	0	1	1	0	1	1	0	0
1	0	1	1	1	0	1	1	0
1	1	0	0	0	0	0	1	1
1	1	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0	0

Output $Z = 1$ only when the pattern is found in the sequence

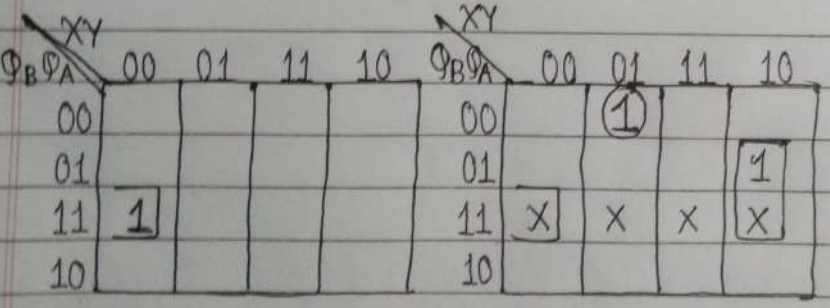
$$Z = \Sigma m(24) + \Sigma\phi(28, 29, 30, 31)$$
$$= Q_C Q_B \bar{Q}_A \bar{X} \bar{Y} + Q_C Q_B Q_A X Y + \phi(29, 30, 31)$$
$$= Q_C Q_B \bar{X} \bar{Y}$$



$$D_A = \overline{Q_B}\,\overline{X}\,\overline{Y} + \overline{Q_A}\,\overline{X}\,\overline{Y} + \overline{Q_C}\,\overline{Q_B}\,\overline{Q_A}\,\overline{X} + Q_C Q_A XY + \overline{Q_C} Q_B Q_A XY$$

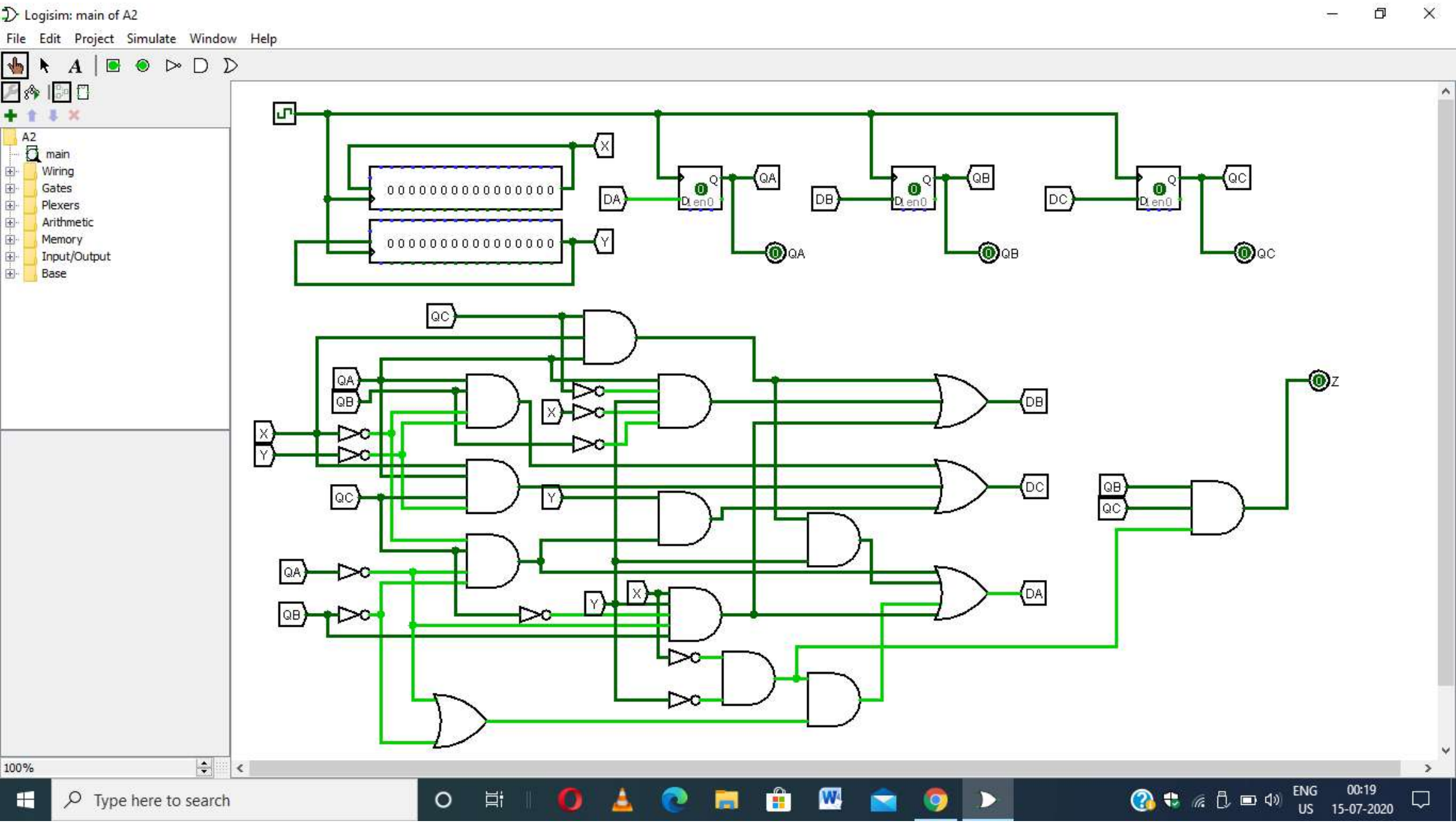


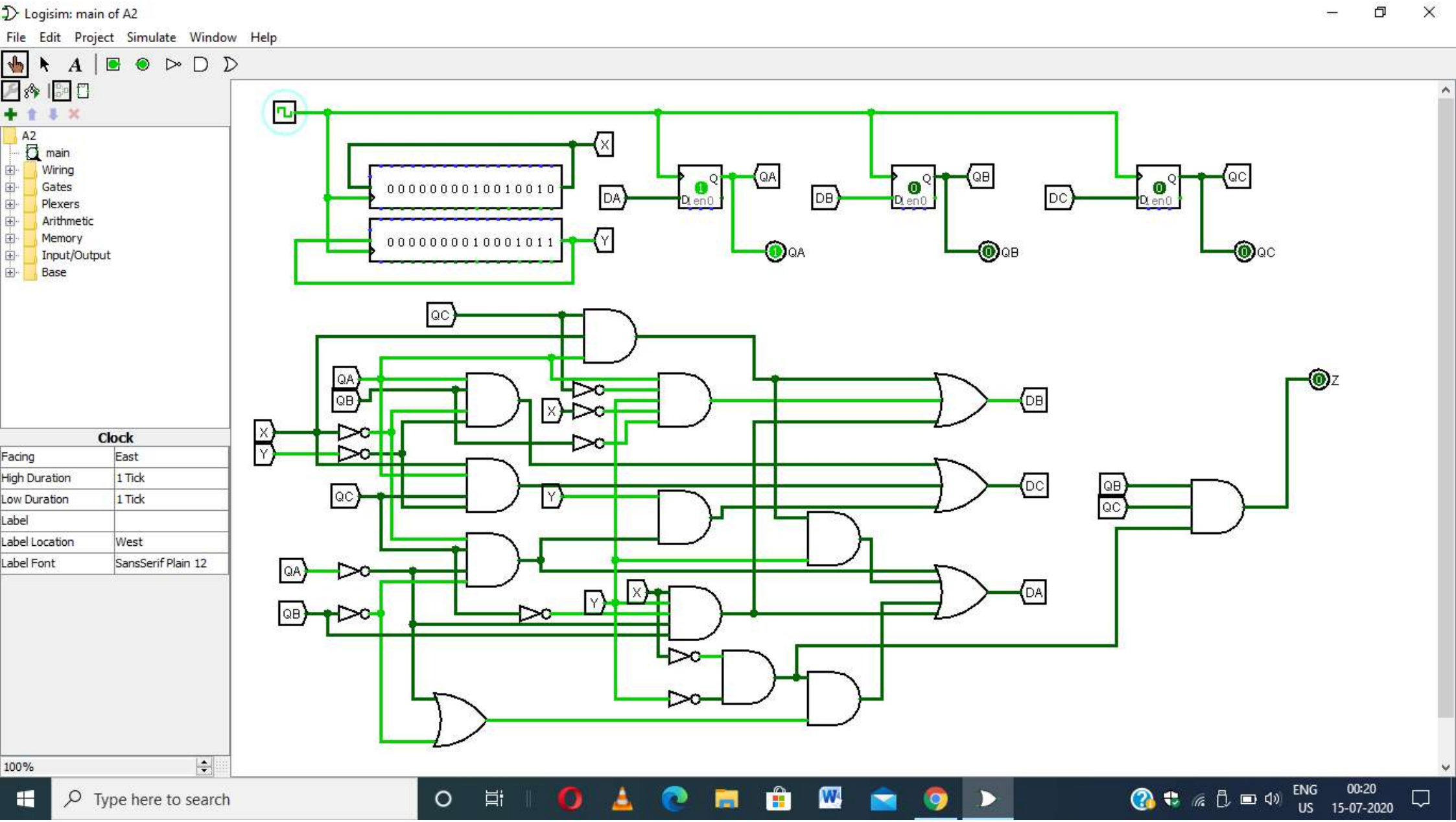
$$D_B = \overline{Q_C}\,\overline{Q_B}\,Q_A\,\overline{X}\,Y + \overline{Q_C}\,Q_B\,\overline{Q_A}\,XY + Q_C Q_A X$$

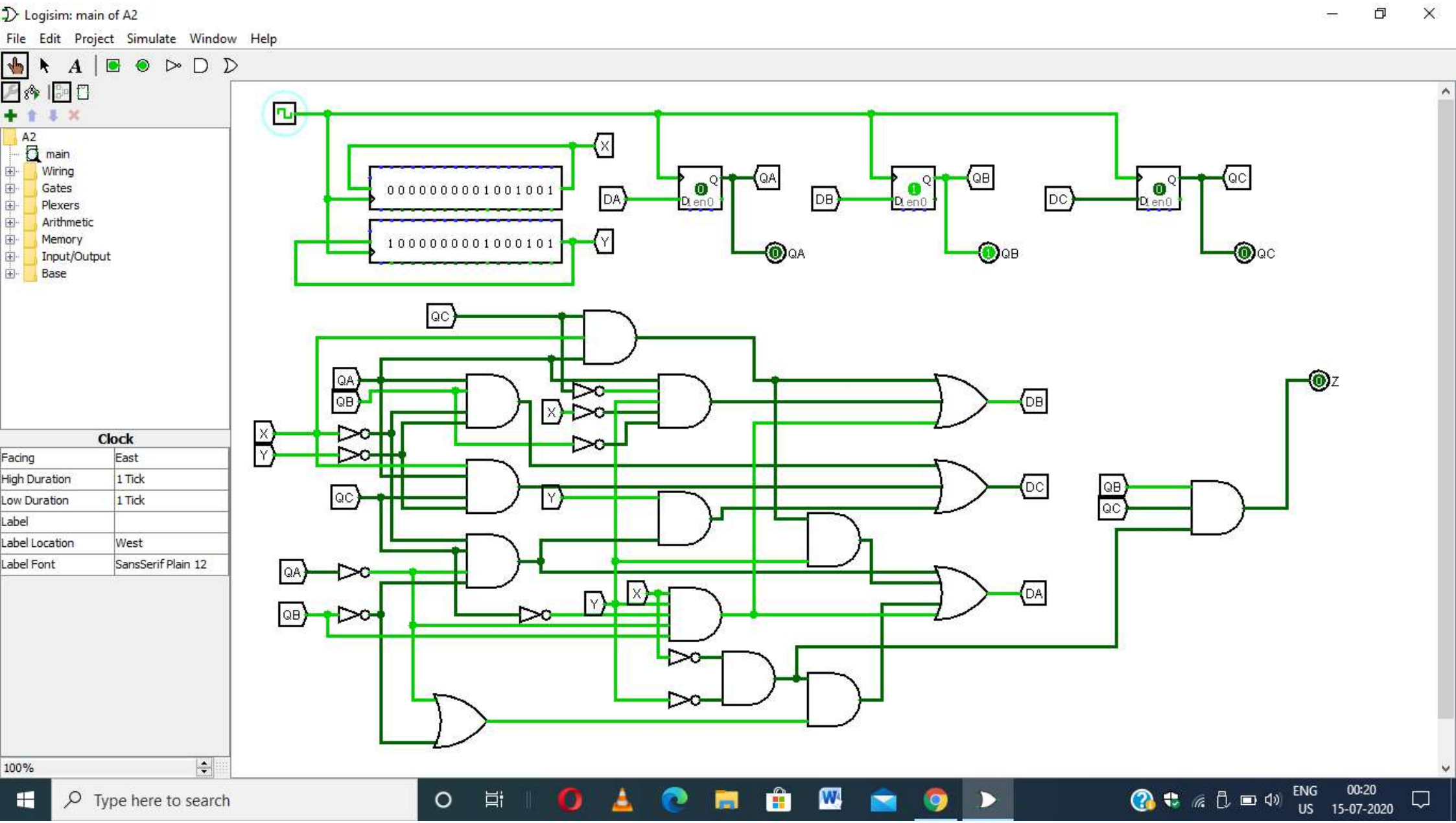


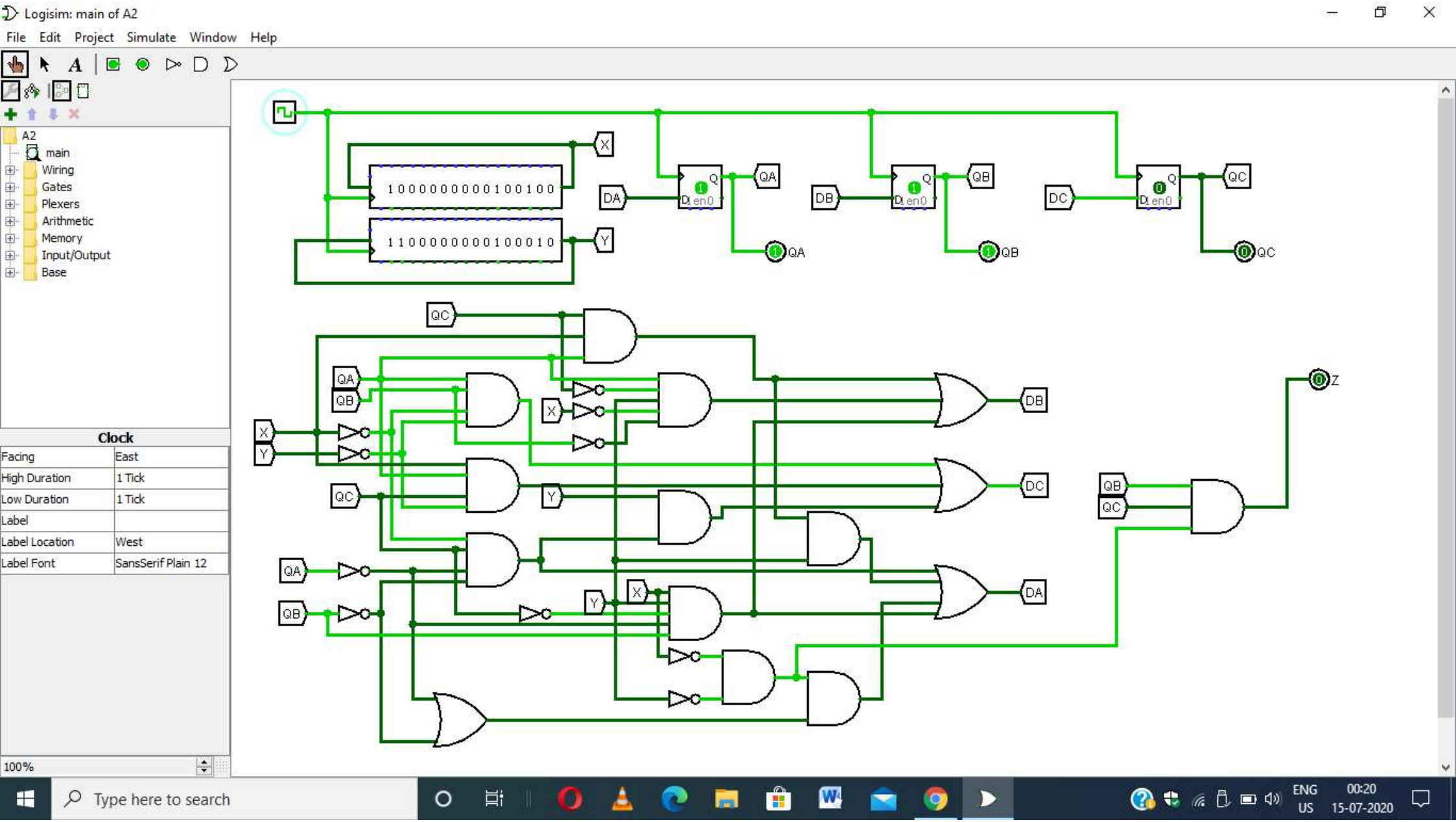
$$D_C = Q_B Q_A \overline{X}\,\overline{Y} + Q_C Q_A X\overline{Y} + Q_C \overline{Q_B}\,\overline{Q_A}\,\overline{X}\,Y$$

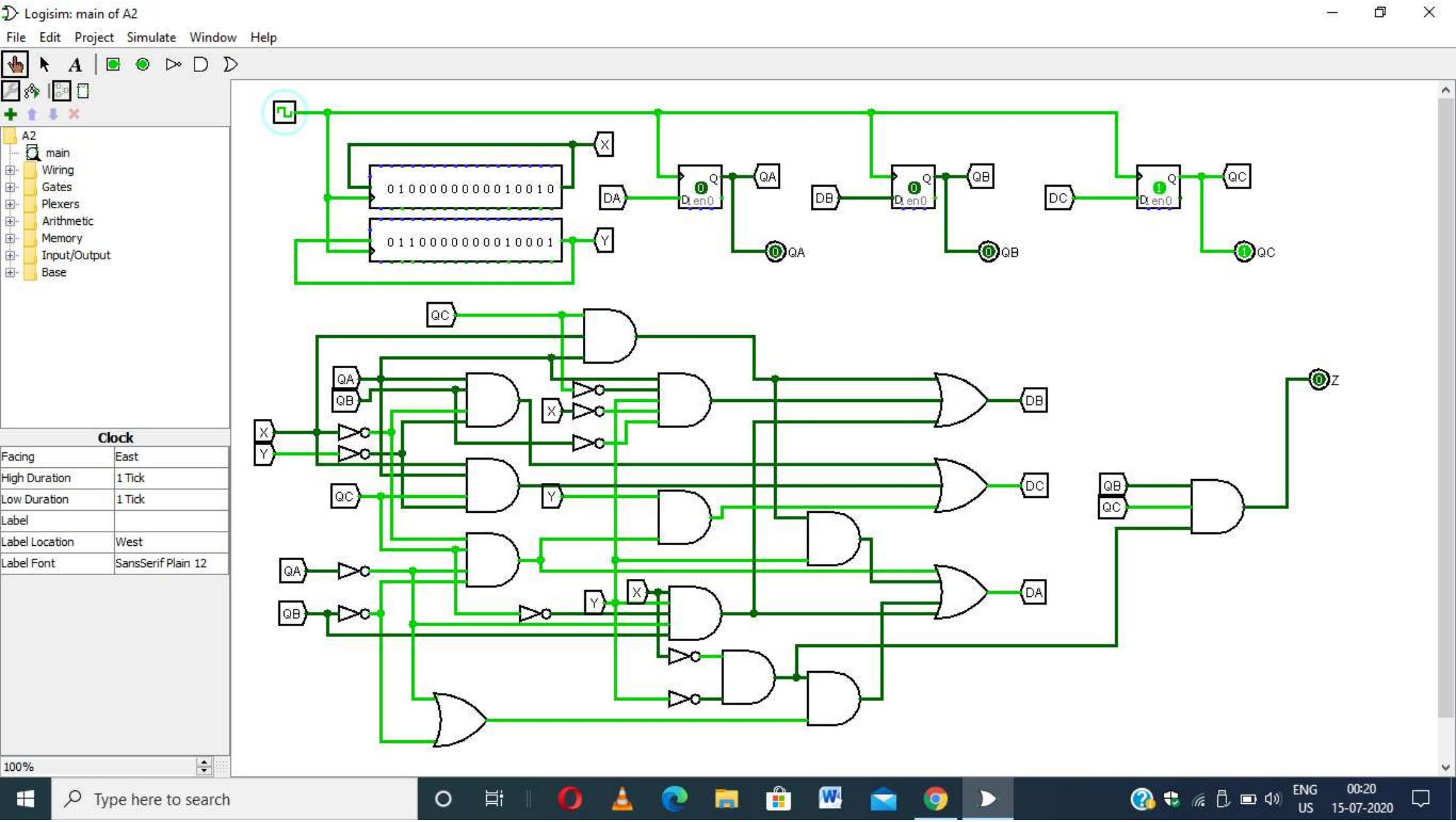
State	Next State/Output			
	X=0 Y=0	X=0 Y=1	X=1 Y=0	X=1 Y=1
S0	S1/0	S0/0	S0/0	S0/0
S1	S1/0	S2/0	S0/0	S0/0
S2	S1/0	S0/0	S3/0	S3/0
S3	S4/0	S0/0	S0/0	S0/0
S4	S1/0	S5/0	S0/0	S0/0
S5	S1/0	S0/0	S6/0	S3/0
S6	S1/1	S0/0	S0/0	S0/0

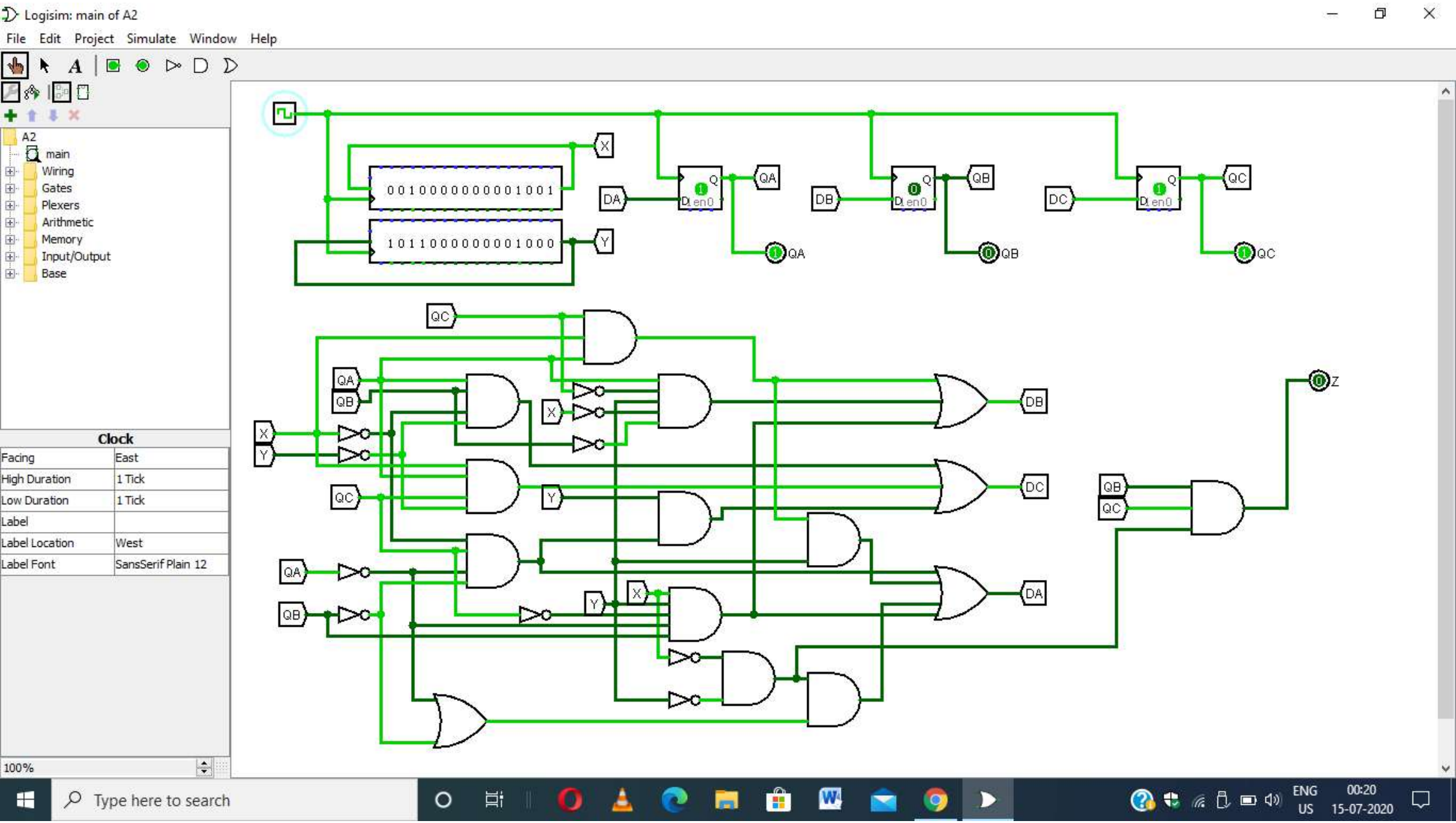

Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
100%
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:19
15-07-2020

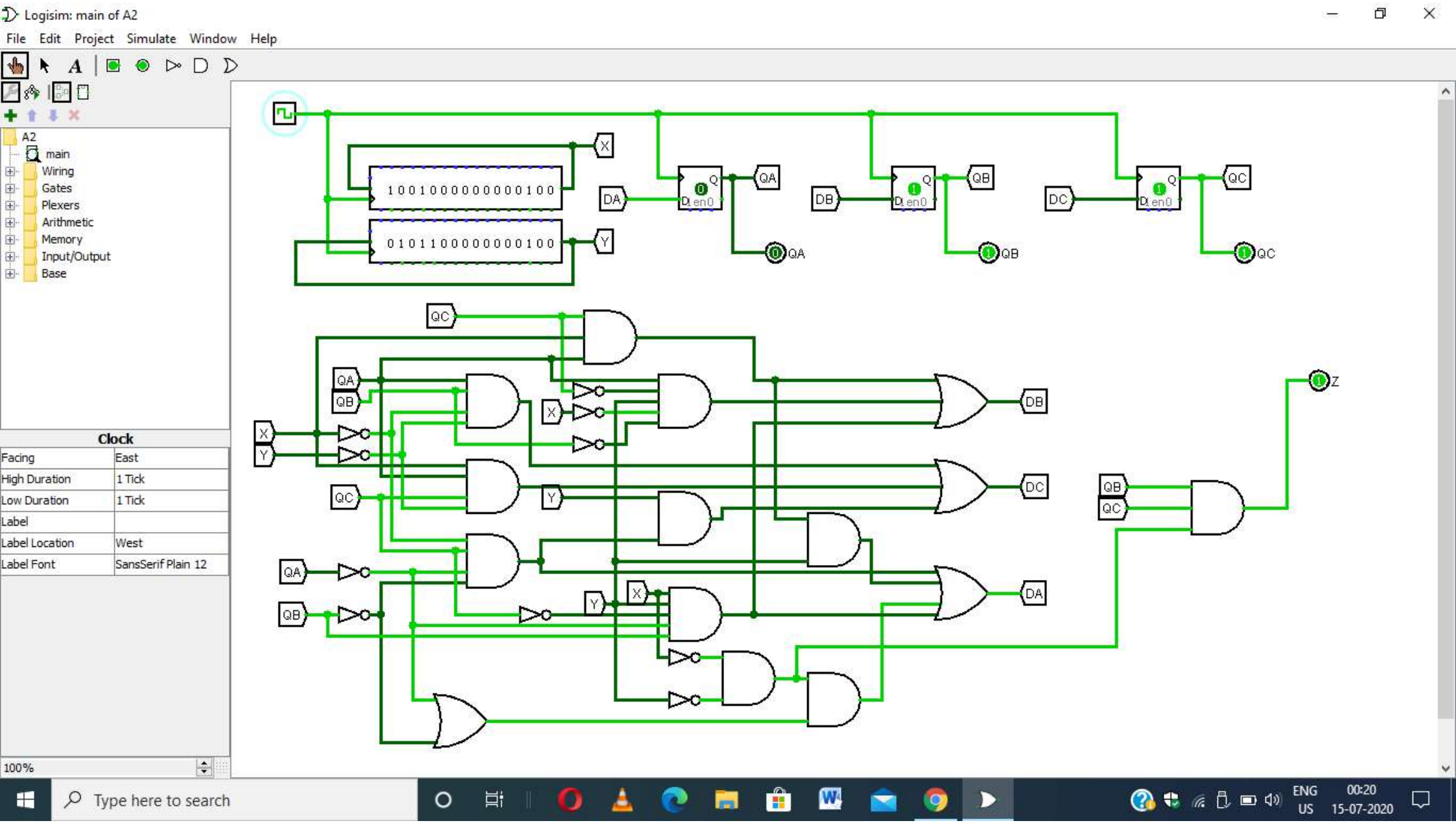

Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0000000010010010
0000000010001011
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020


Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0000000001001001
1000000001000101
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020


Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1000000000100100
1100000000100010
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020


Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020


Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1
1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020


Logisim: main of A2
File Edit Project Simulate Window Help
A2
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0
X
Y
DA
DB
DC
QA
QB
QC
Z
Type here to search
ENG
US
00:20
15-07-2020

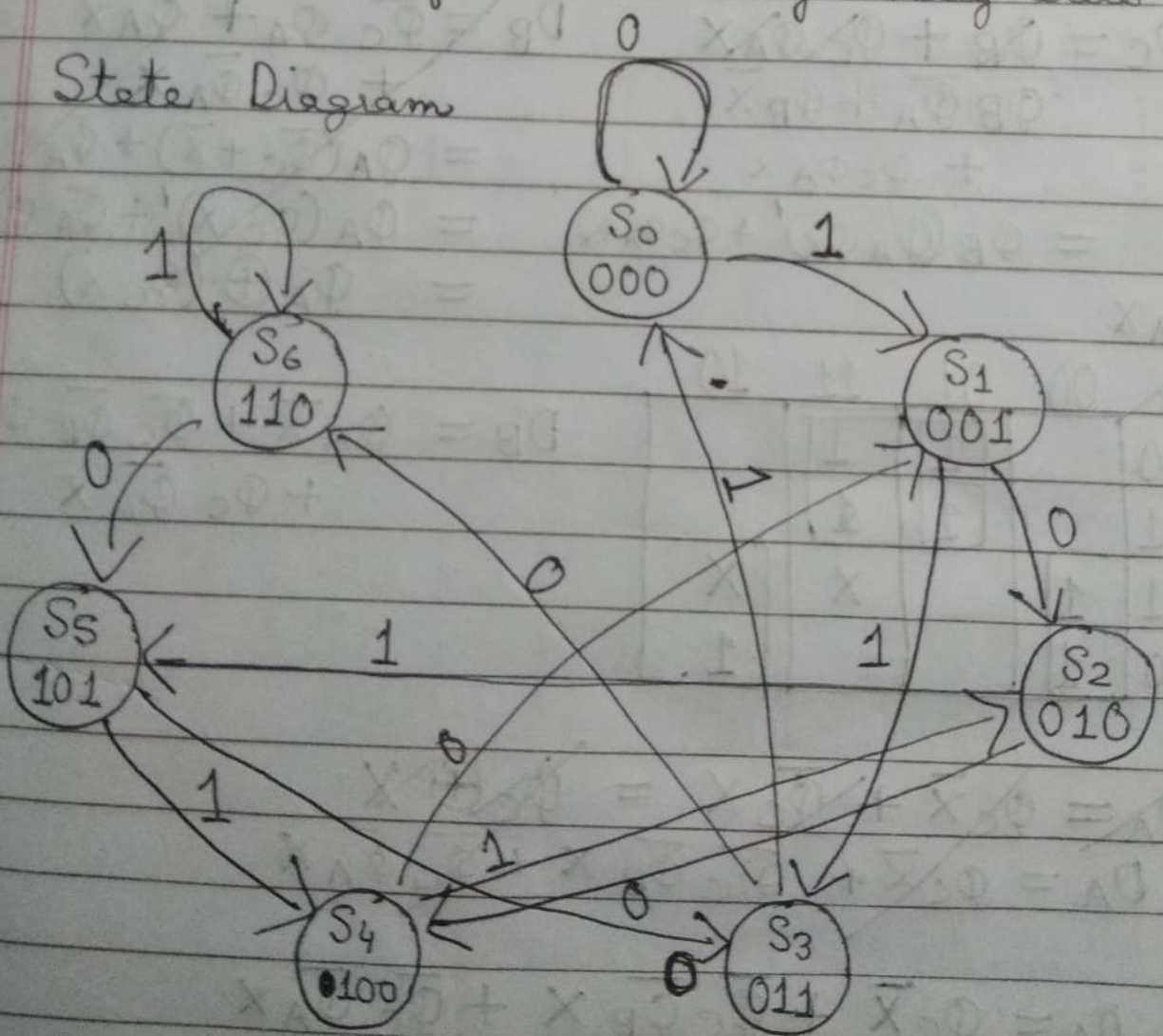
3. Aim of the Experiment :- Given the binary sequence of number (MSB first), Find the remainder of the sequence when divided by 7

Method of Solving :- We will use FSM to solve this problem. Along with FSM, decoder is needed to have any 1 out of 7 output to be high.

Requirements :- 3X8 Decoder, D Flip Flop, Clock

Implementation of Solution using Mealy State Machine

State Diagram

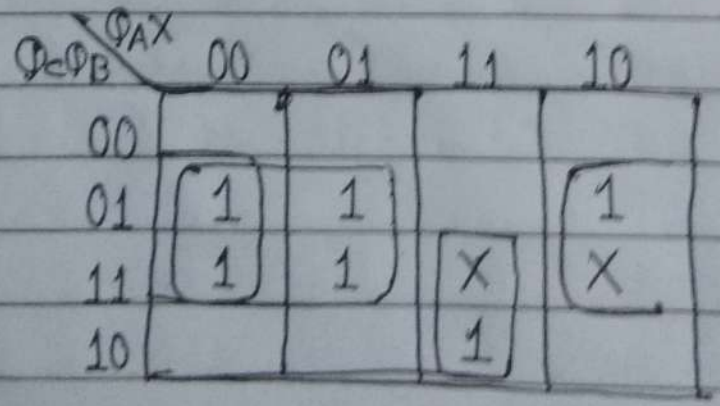


Defining Different States

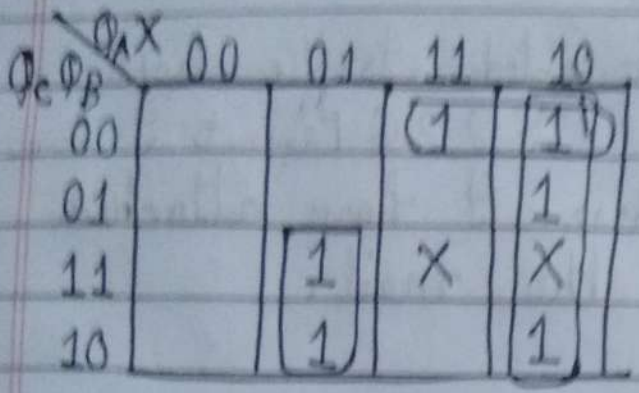
State	Description	Decoder Output 0 1 2 3 4 5 6
S0 (000)	Remainder is 0 R0 LED is ON	1 0 0 0 0 0 0
S1 (001)	Remainder is 1 R1 LED is ON	0 1 0 0 0 0 0
S2 (010)	Remainder is 2 R2 LED is ON	0 0 1 0 0 0 0
S3 (011)	Remainder is 3 R3 LED is ON	0 0 0 1 0 0 0
S4 (100)	Remainder is 4 R4 LED is ON	0 0 0 0 1 0 0
S5 (101)	Remainder is 5 R5 LED is ON	0 0 0 0 0 1 0
S6 (110)	Remainder is 6 R6 LED is ON	0 0 0 0 0 0 1

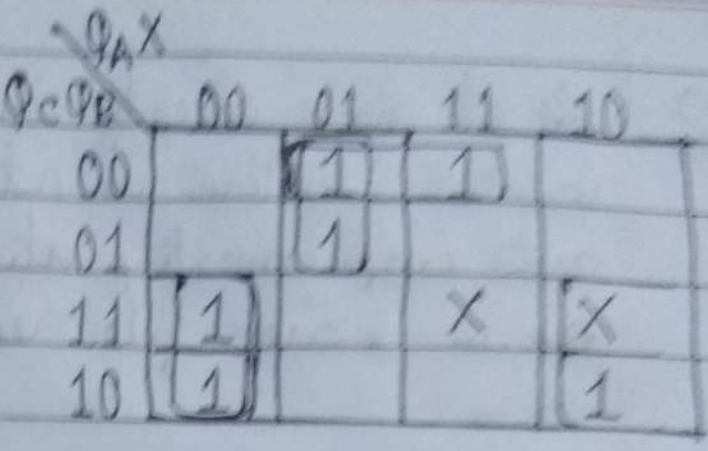
Truth Table

Q_C	Q_B	Q_A	X	Q_C^+ (D_C)	Q_B^+ (D_B)	Q_A^+ (D_A)	Output (R) 0 1 2 3 4 5 6
0	0	0	0	0	0	0	1 0 0 0 0 0 0
0	0	0	1	0	0	1	0 1 0 0 0 0 0
0	0	1	0	0	1	0	0 0 1 0 0 0 0
0	0	1	1	0	1	1	0 0 0 1 0 0 0
0	1	0	0	1	0	0	0 0 0 0 1 0 0
0	1	0	1	1	0	1	0 0 0 0 0 1 0
0	1	1	0	1	1	0	0 0 0 0 0 0 1
0	1	1	1	0	0	0	1 0 0 0 0 0 0
1	0	0	0	0	0	1	0 1 0 0 0 0 0
1	0	0	1	0	1	0	0 0 1 0 0 0 0
1	0	1	0	0	1	1	0 0 0 1 0 0 0
1	0	1	1	1	0	0	0 0 0 0 1 0 0
1	1	0	0	1	0	1	0 0 0 0 0 1 0
1	1	0	1	1	1	0	0 0 0 0 0 0 1



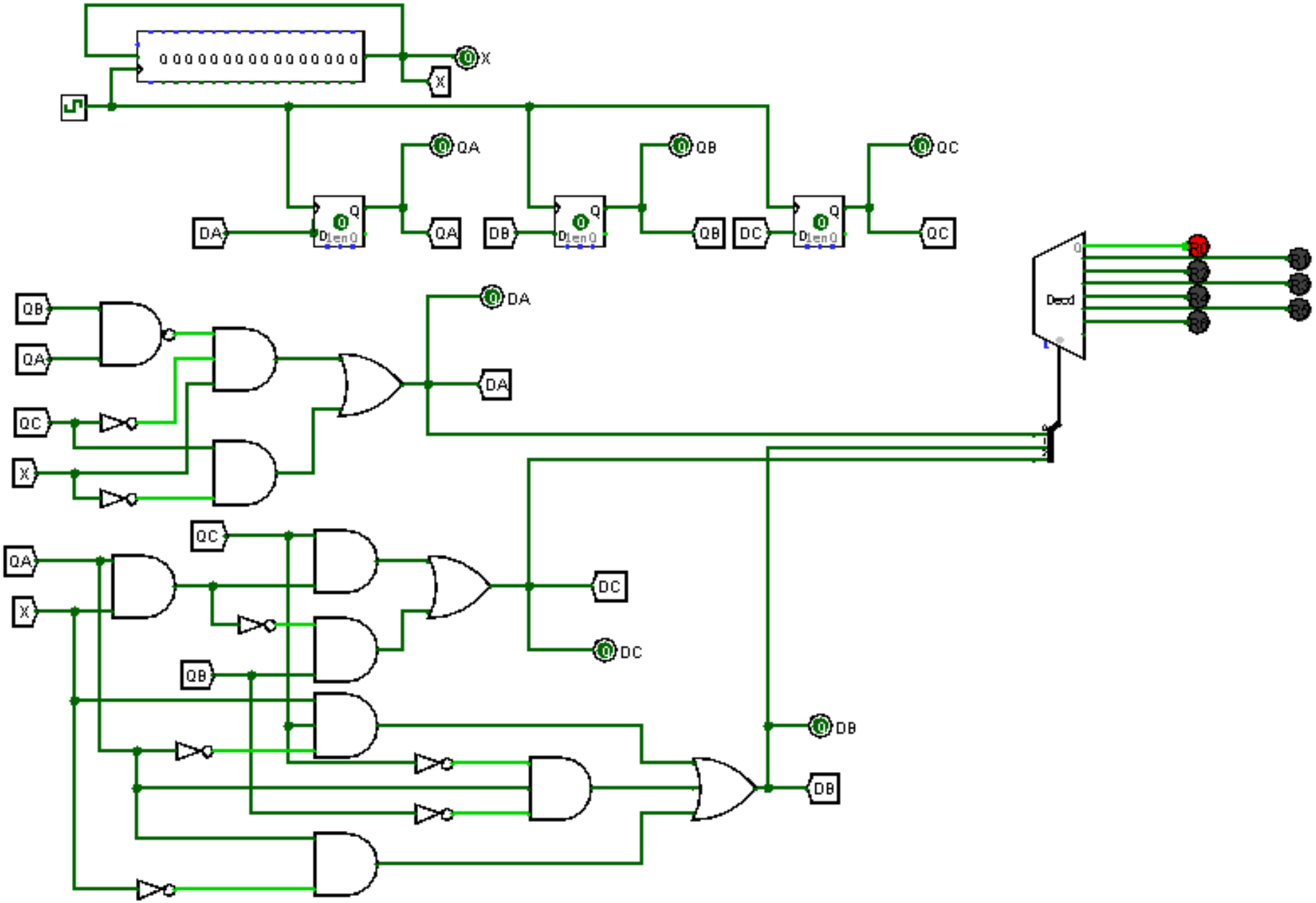
$$D_C = Q_B\overline{Q}_A + Q_B\overline{X} + Q_C Q_A \odot X$$
$$= Q_B(Q_A \odot X)' + Q_C(Q_A X)$$

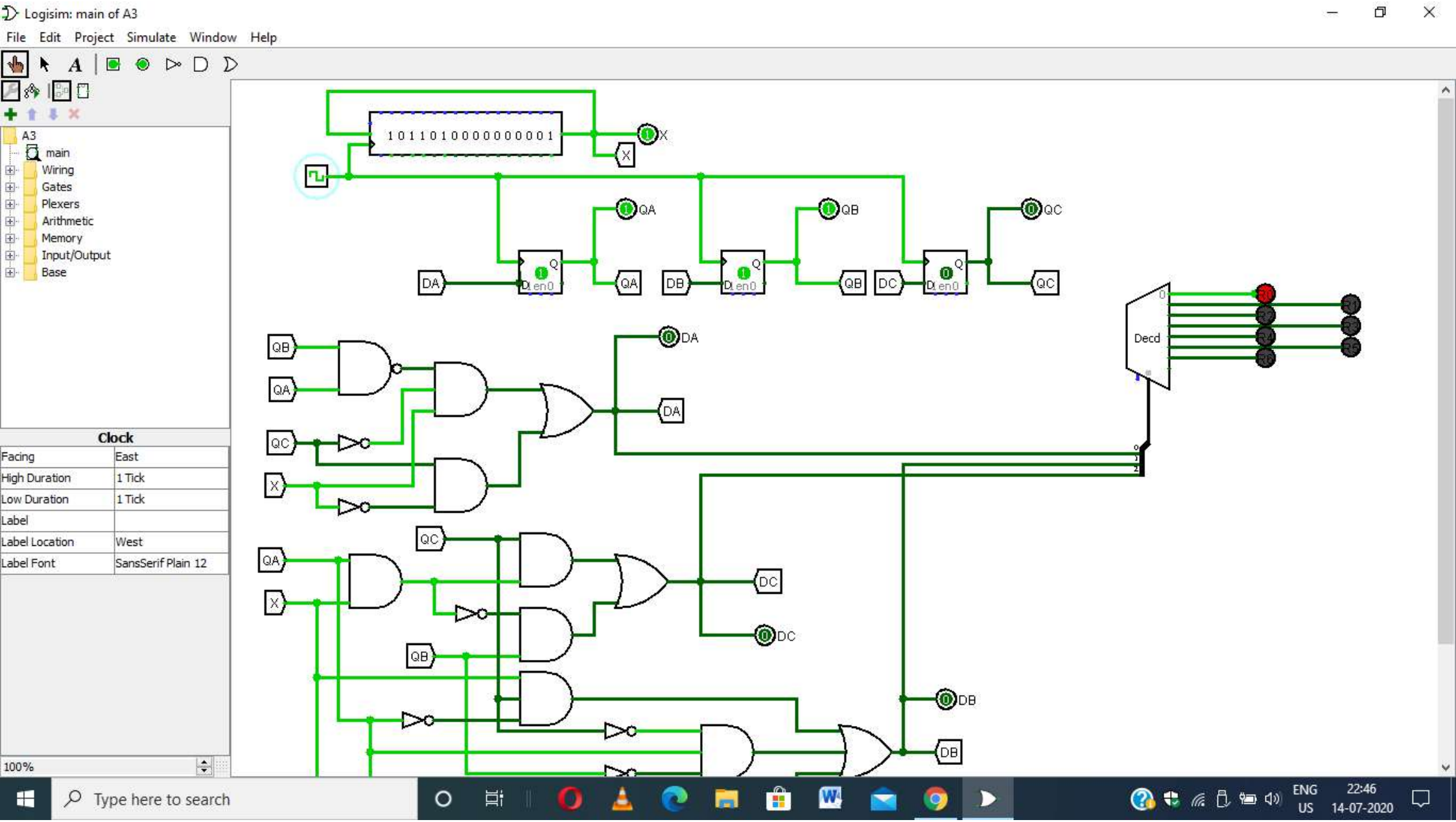


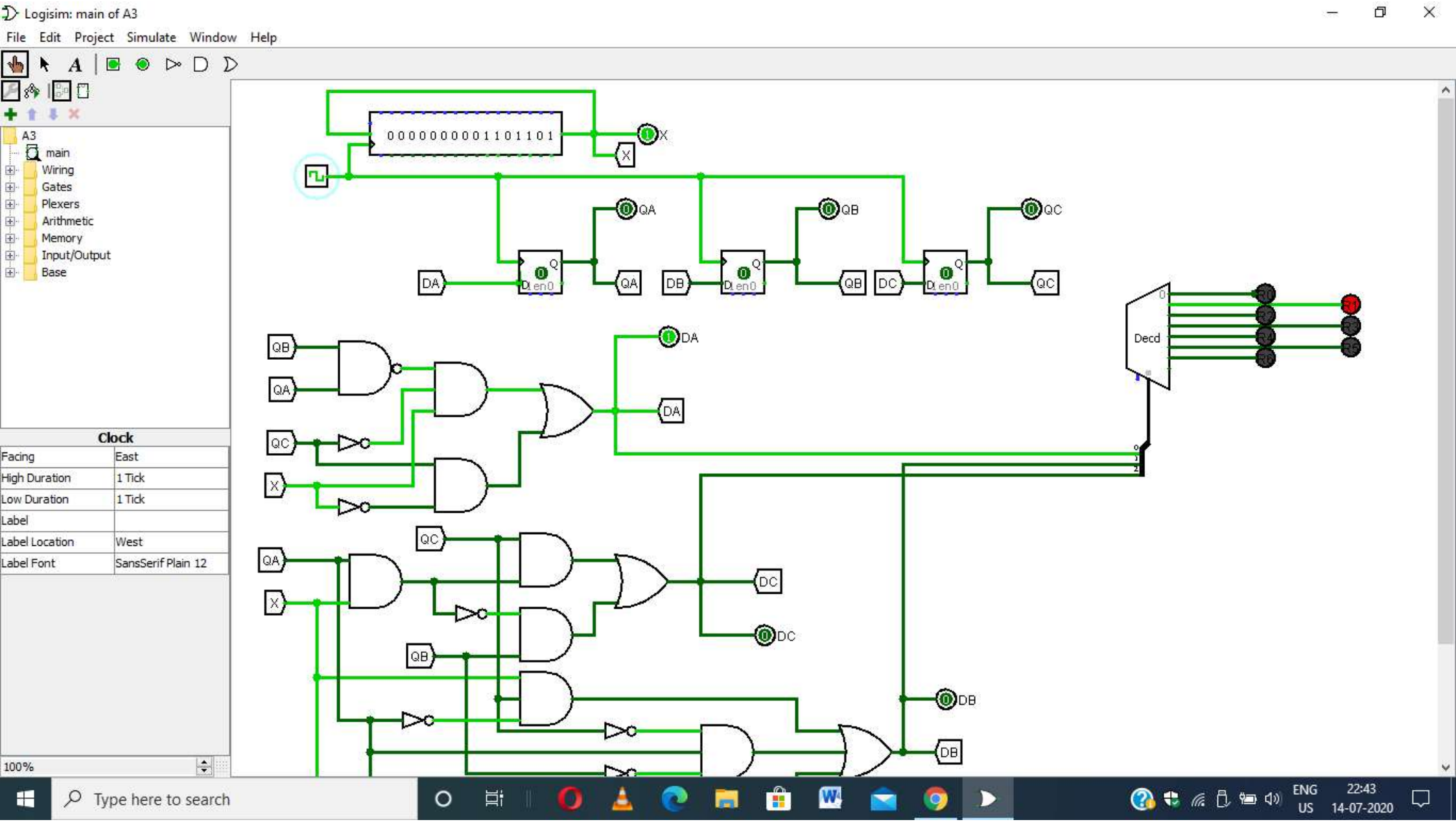


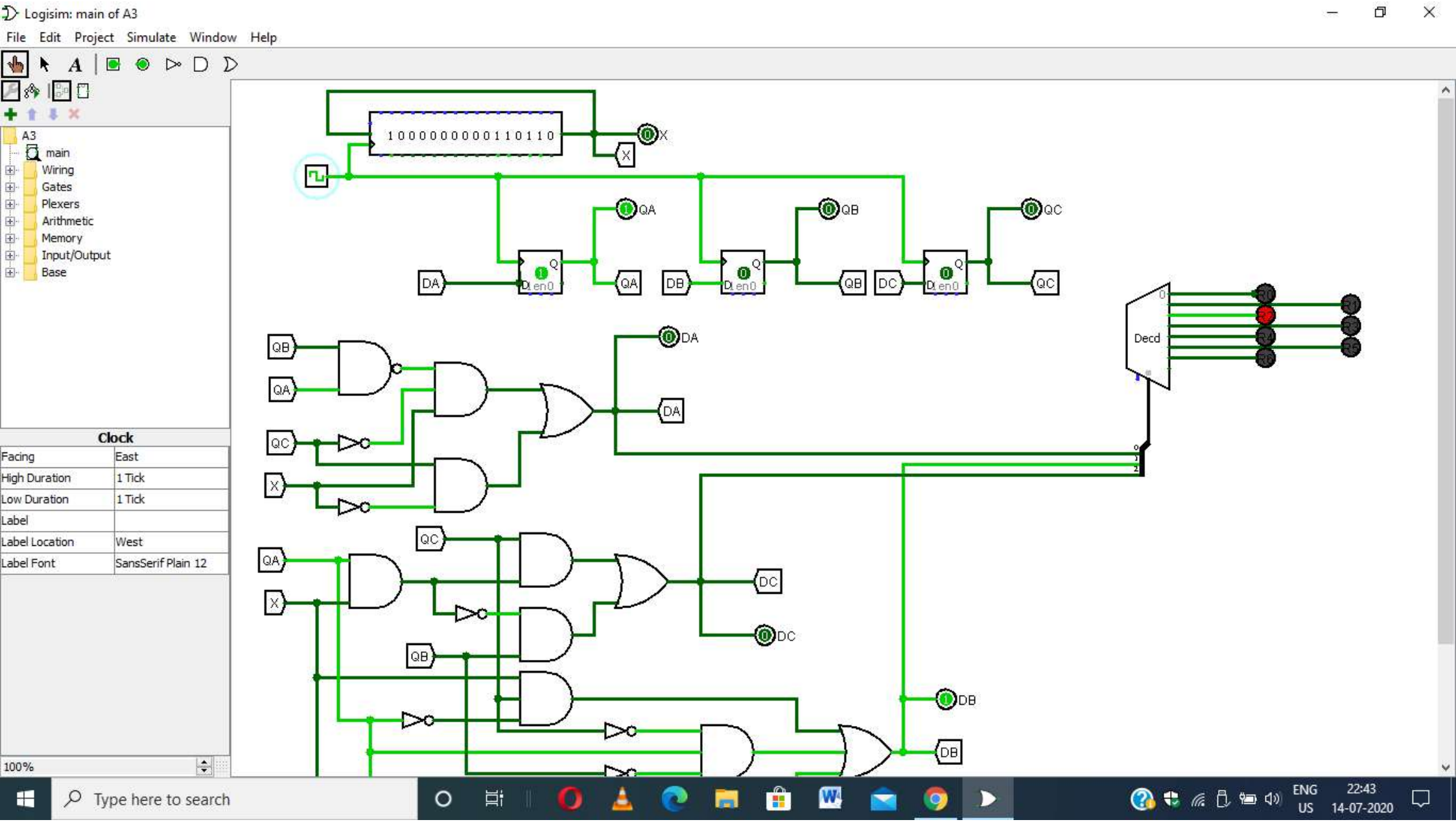
$$D_B = \bar{Q}_C \bar{Q}_B Q_A + Q_A \bar{X} + Q_C \bar{Q}_A X$$

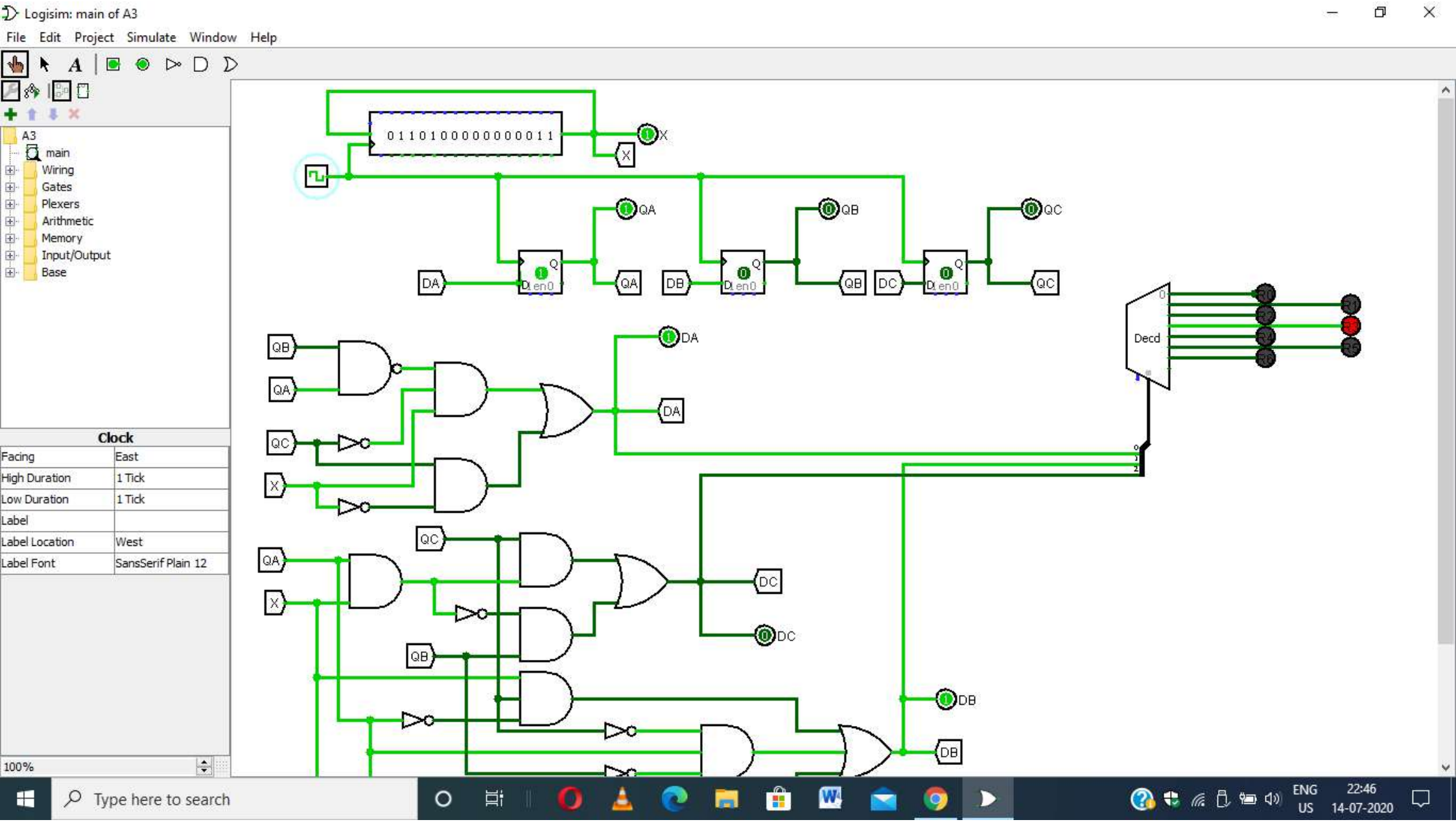
$$D_A = Q_C \bar{X} + \bar{Q}_C \bar{Q}_B X + \bar{Q}_C \bar{Q}_A$$
$$= Q_C \bar{X} + \bar{Q}_C X (Q_B Q_A)'$$

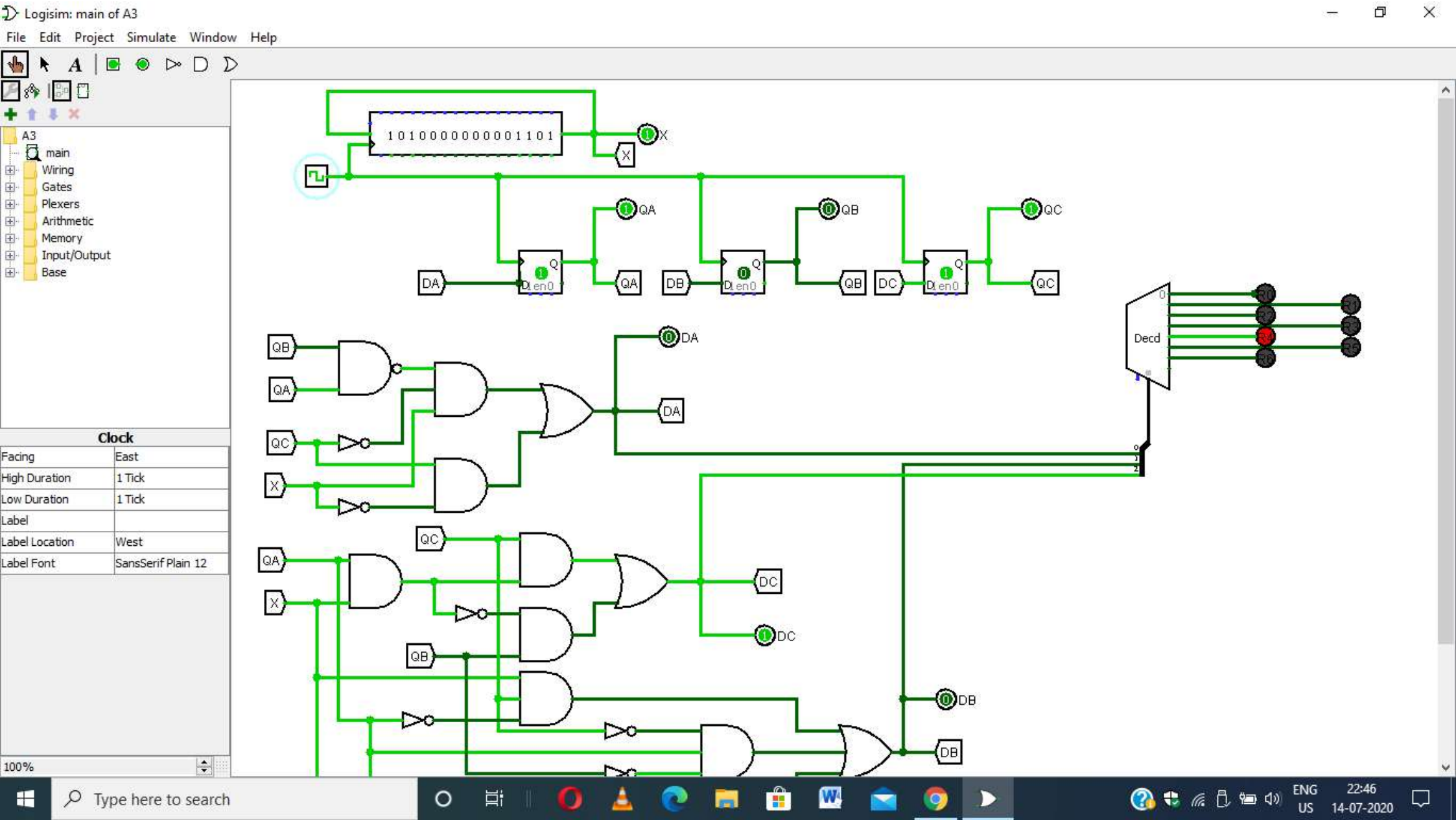

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
X
X
QA
QB
QC
DA
QA
DB
QB
DC
QC
Q
D Len Q
Decd
QB
QA
QC
X
DA
DA
QC
QA
X
QB
DC
DC
DB
DB

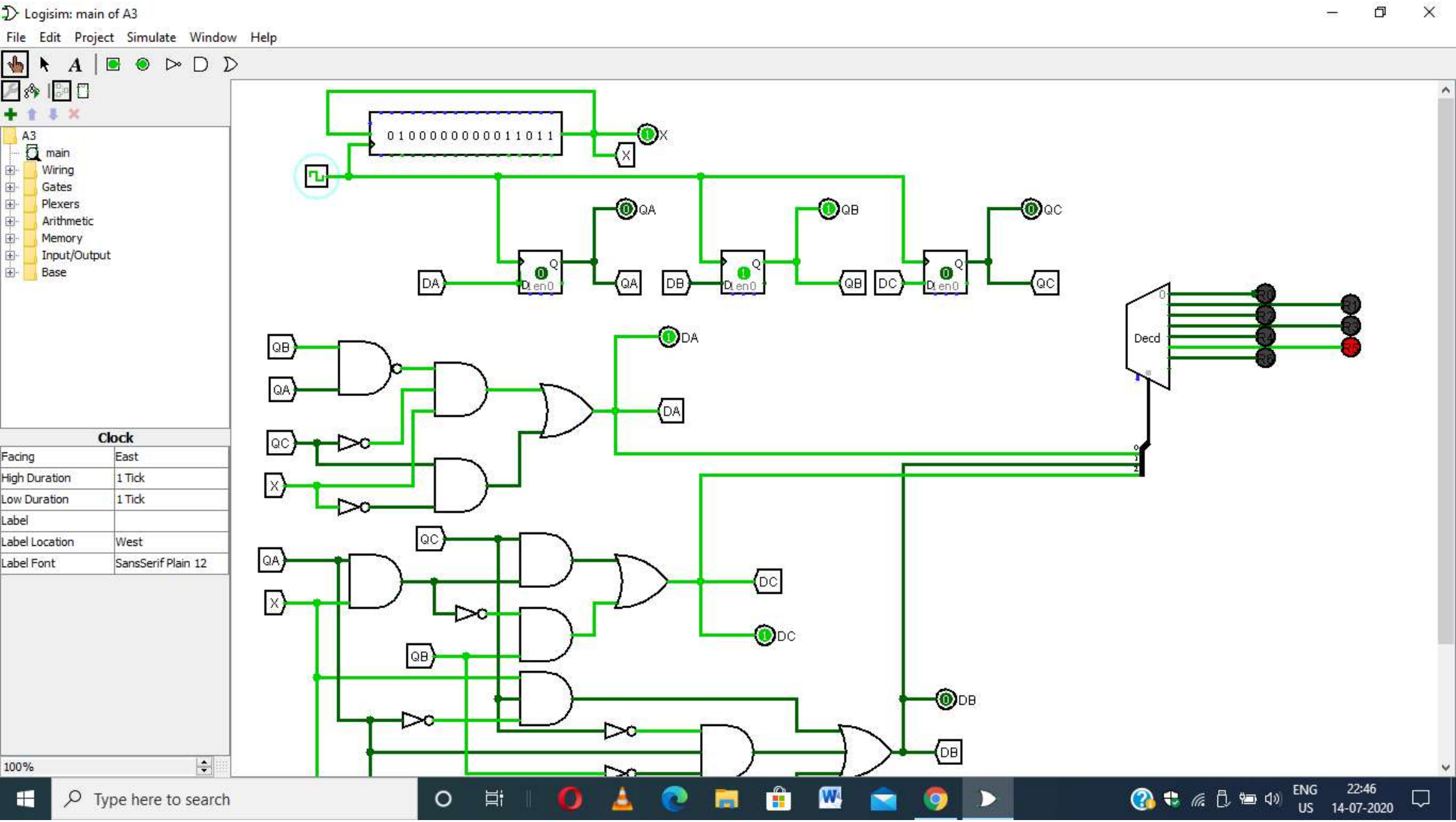

Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1 0 1 1 0 1 0 0 0 0 0 0 0 0 1
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:46
14-07-2020

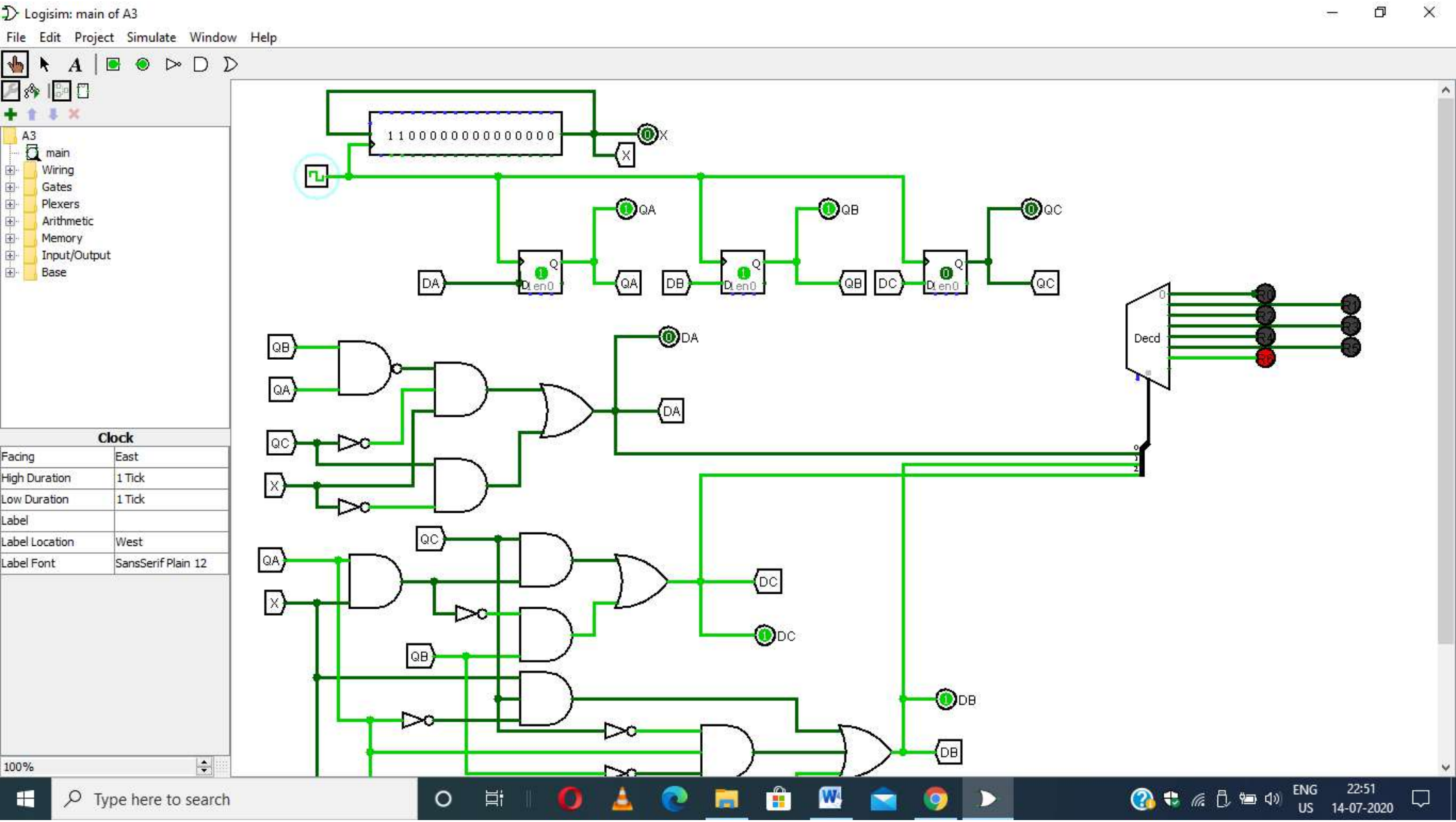

Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0000000001101101
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:43
14-07-2020


Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1000000000110110
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:43
14-07-2020


Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:46
14-07-2020


Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:46
14-07-2020


Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:46
14-07-2020


Logisim: main of A3
File Edit Project Simulate Window Help
A3
main
Wiring
Gates
Plexers
Arithmetic
Memory
Input/Output
Base
Clock
Facing East
High Duration 1 Tick
Low Duration 1 Tick
Label
Label Location West
Label Font SansSerif Plain 12
100%
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
X
QA
QB
QC
DA
DB
DC
Decd
Type here to search
ENG
US
22:51
14-07-2020