



LAPORAN PRAKTIKUM

IF310303

PRAKTIKUM SISTEM DIGITAL

MODUL: 6

MULTIPLEKSER DAN DEMULTIPLEKSER

NAMA : Muhammad Alwiza Ansyar
NIM : M0520051
HARI : Jumat
TANGGAL : 20 November 2020
WAKTU : 10.15 – 11.05 WIB
ASISTEN : Akhtar Bariq Rahman

PROGRAM STUDI INFORMATIKA

UNIVERSITAS SEBELAS MARET

2020

Modul 6

MULTIPLEKSER DAN DEMULTIPLEKSER

Muhammad Alwiza Ansyar (M0520051) / 20 November 2020

Email : alwiza21@student.uns.ac.id

Asisten : Akhtar Bariq Rahman

Abstraksi— Berikut merupakan laporan praktikum untuk modul 6 yang memiliki fokus bahasan tentang multiplexer dan demultiplexer. Multiplexer dan demultiplexer adalah salah dua rangkaian logika kombinasional. Multiplexer berfungsi untuk menyeleksi *data input*, sedangkan demultiplexer berfungsi untuk mendistribusikan *data input*. Pada laporan praktikum ini, akan dibahas tentang pengekspresian 4x1 Multiplexer dan 1x4 Demultiplexer dalam rangkaian logika serta tabel kebenarannya

Kata kunci— Multiplexer, demultiplexer

I. PENDAHULUAN

Dalam kehidupan sehari-hari, kita dihadapkan oleh beragam informasi yang bercampur antara *hoax* atau fakta, penting atau tidak penting, serta baru atau lama. Tidak semua informasi harus kita terima dan tidak semua informasi bisa disebarkan lagi ke orang lain. Begitu juga pada rangkaian logika. Tidak semua keadaan logika adalah penting/diperlukan pada keadaan tertentu, dan tidak semua *output* memerlukan informasi keadaan logika dari suatu *input*.

Maka dari itu, terdapat rangkaian kombinasional untuk menangani masalah diatas, yaitu multiplexer dan demultiplexer. Multiplexer akan menyeleksi *data input* mana yang akan diteruskan ke *output*, sedangkan demultiplexer akan mendistribusikan *data input* ke *output* yang terpilih. Keduanya memiliki fungsi yang saling berkebalikan.

II. DASAR TEORI

2.1 Multiplexer

Multiplexer adalah suatu rangkaian kombinasional yang menyeleksi *data input* untuk diteruskan pada *output* menggunakan *selector* (yang juga merupakan *input*). Rangkaian ini sering disebut data *selector* dan disingkat mux. Mux memiliki dua jenis *input*, yaitu *data input* dan *selector input*. *Selector input* akan memilih *data input* mana yang akan diteruskan ke *output*, sedangkan *data input* sendiri membawa informasi yang akan diteruskan (seperti apakah berupa low logic atau high logic). Kondisi logika dari *data input* yang terpilih oleh *selector* akan ditampilkan pada *output*.

Mux memiliki ciri khas yaitu hanya memiliki satu *output*. Pada beberapa mux, terdapat *input* tambahan berupa enabler *input*, yaitu *input* yang menentukan apakah mux beroperasi atau tidak beroperasi (seperti tombol on/off). Beberapa jenis mux antara lain 2x1 Multiplexer, 4x1 Multiplexer, 8x1 Multiplexer, dan 16x1 Multiplexer.

Terdapat hubungan antara jumlah *data input* dan *selector input*, yaitu $n = 2^s$, dengan n adalah jumlah *data input* dan s adalah jumlah *selector input*. Maka dari itu, jumlah *data input* dari multiplexer selalu merupakan hasil dari perpangkatan

2.2 Demultiplexer

Demultiplexer adalah suatu rangkaian kombinasional yang mendistribusikan *data input* untuk diteruskan pada *output* yang diinginkan menggunakan *selector* (yang juga merupakan *input*). Rangkaian ini sering disebut data distributor dan disingkat demux. Demux memiliki dua jenis *input*, yaitu *data input* dan *selector input*. *Selector input* akan memilih *output* mana yang akan menerima informasi dari *data input*, sedangkan *data input* sendiri membawa informasi yang akan diteruskan (seperti apakah berupa low logic atau high logic). *Output* yang terpilih oleh *selector* akan menampilkan kondisi logika dari *data input*.

Demux memiliki ciri khas yaitu hanya memiliki satu *data input*. Pada beberapa demux, juga terdapat *input* tambahan berupa enabler *input*, yaitu *input* yang menentukan apakah demux beroperasi atau tidak beroperasi (seperti tombol on/off). Beberapa jenis demux antara lain 1x2 Demultiplexer, 1x4 Demultiplexer, 1x8 Demultiplexer, dan 1x16 Demultiplexer.

Terdapat hubungan antara jumlah *data input* dan *selector input*, yaitu $m = 2^s$ dengan m adalah jumlah *output* dan s adalah jumlah *selector input*. Maka dari itu, jumlah *output* dari demultiplexer selalu merupakan hasil dari perpangkatan 2.

III. ALAT DAN LANGKAH PERCOBAAN

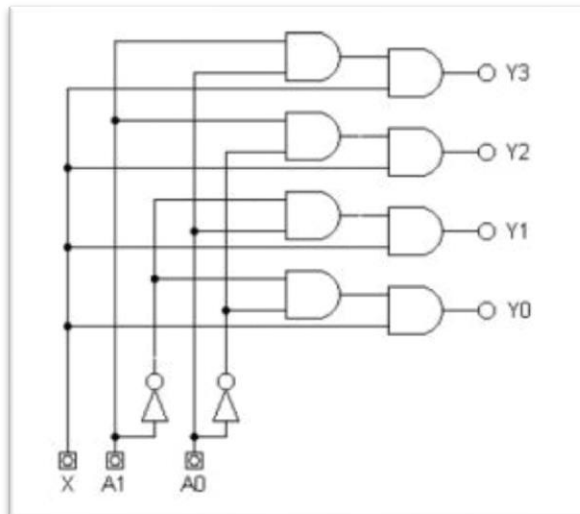
3.1 Alat

1. PC/Laptop
2. Aplikasi Digital Works

3.2 Langkah Percobaan

Demultiplexer

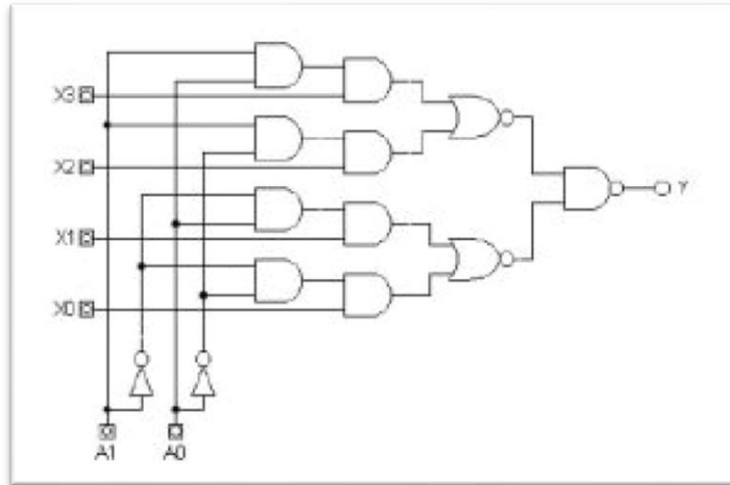
1. Buka Aplikasi Digital Works
2. Buatlah rangkaian demultiplexer 1x4 berikut



3. Amati dan catat *output* terhadap kombinasi keadaan *input*
4. Buatlah rangkaian demultiplexer 1x4 menggunakan IC 74155
5. Amati dan catat *output* terhadap kombinasi keadaan *input*

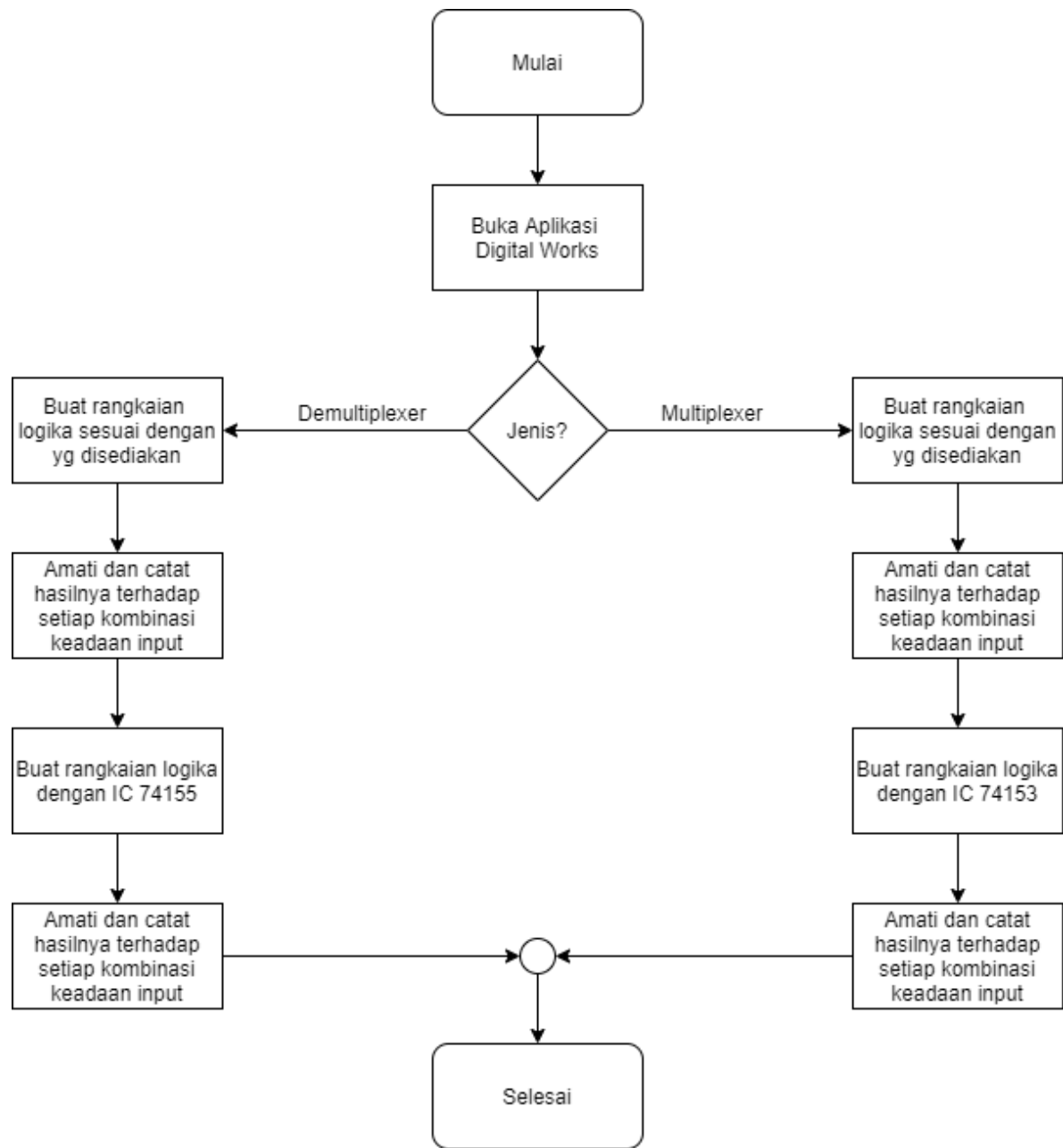
Multiplexer

6. Buatlah rangkaian multiplexer 4x1 berikut



7. Amati dan catat *output* terhadap kombinasi keadaan *input*
8. Buatlah rangkaian multiplexer 4x1 menggunakan IC 74153
9. Amati dan catat *output* terhadap kombinasi keadaan *input*

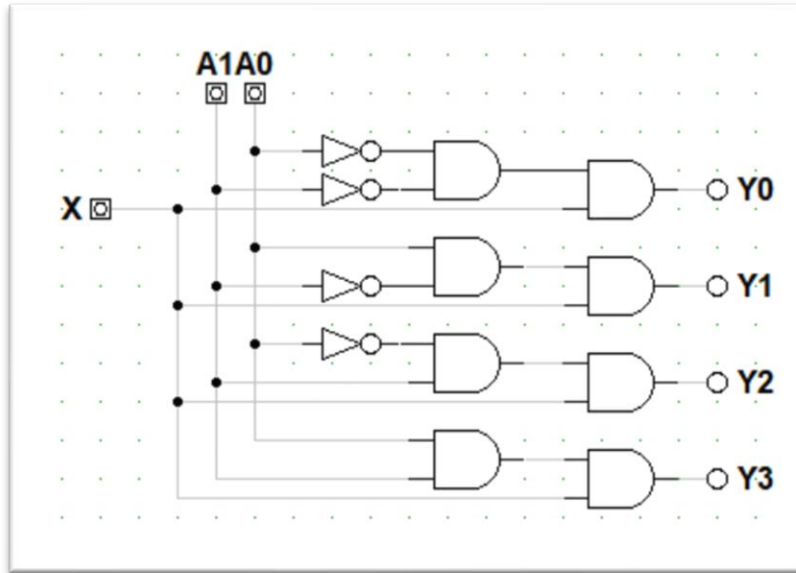
Diagram alur



IV. HASIL DAN ANALISIS PERCOBAAN

4.1 Demultiplexer 1x4

Rangkaian logika menggunakan gerbang logika dasar:



Tabel kebenaran:

A1	A0	X	Y0	Y1	Y2	Y3
0	0	0	0	x	x	x
0	0	1	1	x	x	x
0	1	0	x	0	x	x
0	1	1	x	1	x	x
1	0	0	x	x	0	x
1	0	1	x	x	1	x
1	1	0	x	x	x	0
1	1	1	x	x	x	1

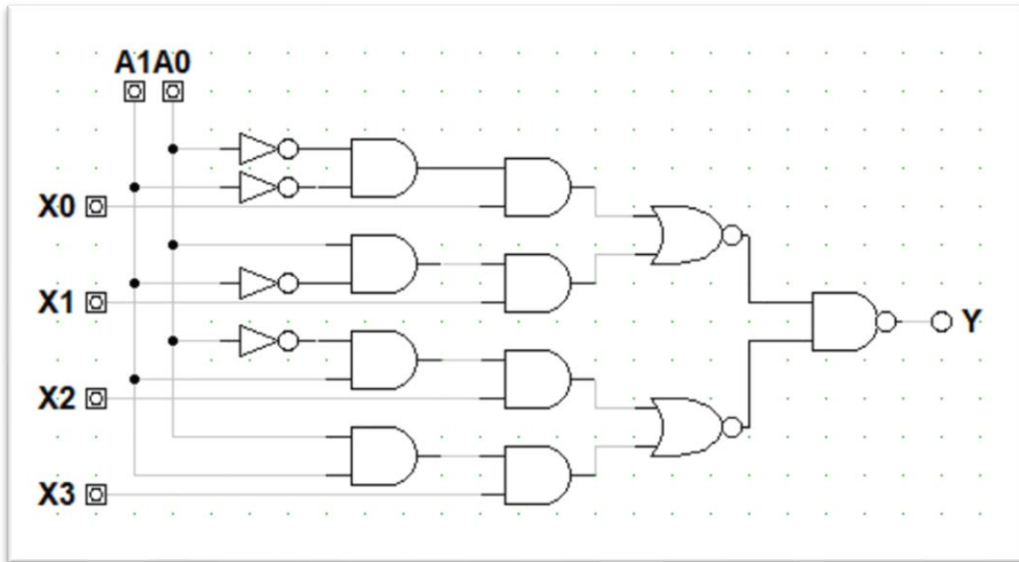
Pada rangkaian ini, A0 dan A1 adalah *selector input*, X adalah *data input*, serta Y0, Y1, Y2, dan Y3 adalah *output*. Ketika suatu *output* terpilih oleh *selector*, maka *output* tersebut akan menampilkan kondisi logika dari *data input*, sedangkan kondisi logika *output* lain yang tidak terpilih adalah tidak dipedulikan/keadaan *don't care* (ditandai dengan x). Pada saat keadaan *don't care*, keadaan logika *output* tersebut secara *default*/bawaan adalah berlogika 0.

Rangkaian logika menggunakan IC 74155:

Tidak ditemukan IC 74155 pada Digital Works sehingga rangkaian ini tidak dibuat.

4.2 Multiplexer 4x1

Rangkaian logika menggunakan gerbang logika dasar:



Tabel kebenaran:

A1	A0	X0	X1	X2	X3	Y
0	0	0	×	×	×	0
0	0	1	×	×	×	1
0	1	×	0	×	×	0
0	1	×	1	×	×	1
1	0	×	×	0	×	0
1	0	×	×	1	×	1
1	1	×	×	×	0	0
1	1	×	×	×	1	1

Pada rangkaian ini, A0 dan A1 adalah *selector input*, X0, X1, X2, dan X3 adalah *data input*, serta Y adalah *output*. Ketika suatu *data input* dipilih oleh *selector*, maka *output* akan menampilkan kondisi logika dari *data input* tersebut, sedangkan kondisi logika *data input* lain yang tidak terpilih adalah tidak dipedulikan/keadaan *don't care* (ditandai dengan ×).

Rangkaian logika menggunakan IC 74153:

Tidak ditemukan IC 74153 pada Digital Works sehingga rangkaian ini tidak dibuat.

V. KESIMPULAN

Multiplexer dan demultiplexer adalah rangkaian kombinasional yang memiliki fungsi saling berkebalikan. Multiplexer berfungsi untuk menyeleksi *data input* untuk diteruskan ke *output*. Demultiplexer berfungsi untuk mendistribusikan *data input* ke *output* yang diinginkan. Multiplexer sering disebut data *selector* dan disingkat mux, serta memiliki ciri khas yaitu hanya memiliki satu *output*. Demultiplexer sering disebut data distributor dan disingkat demux serta memiliki ciri khas yaitu hanya memiliki satu *data input*

Keduanya memiliki dua jenis *input*, yaitu *data input* dan *selector input*. *Data input* berisi informasi yang akan diteruskan, sedangkan *selector input* berfungsi sebagai pemilih/penyeleksi. *Selector input* pada multiplexer akan memilih *data input* yang akan diteruskan ke *output*, sedangkan *selector input* pada demultiplexer akan memilih *output* yang akan menerima informasi dari *data input*. Jumlah dari *data input* atau *output* dengan *selector input* adalah berhubungan. Pada mux, rumusnya ialah $n = 2^s$ dengan n adalah jumlah *data input* dan s adalah jumlah *selector input*. Pada demux, rumusnya ialah $m = 2^s$ dengan m adalah jumlah *output* dan s adalah jumlah *selector input*

Pada praktikum ini, dibuat rangkaian untuk multiplexer 4x1 dan demultiplexer 1x4. Keduanya memiliki fungsi yang saling berkebalikan. Hasil dari simulasi rangkaian terkait pada Digital Works adalah sesuai dengan seharusnya, yang lalu dicatat pada tabel kebenaran.

Pada praktisnya, mungkin saja untuk memiliki jumlah *data input* atau *output* yang bukan merupakan hasil dari perpangkatan 2. Dengan itu, maka tidak semua nilai dari *selector* memiliki arti sehingga diperlukan larangan/*restriction* untuk tidak membolehkan penggunaan *selector* tersebut. Contohnya ialah kondisi dimana hanya terdapat 3 *data input* pada 4x1 Multiplexer. *Data input* ke-4 (n_3) adalah nihil sehingga *selector* yang merujuk ke *data input* tersebut (yaitu kombinasi 11) tidak memiliki arti. Maka, diperlukan *restriction* untuk tidak menggunakan *selector* tersebut.

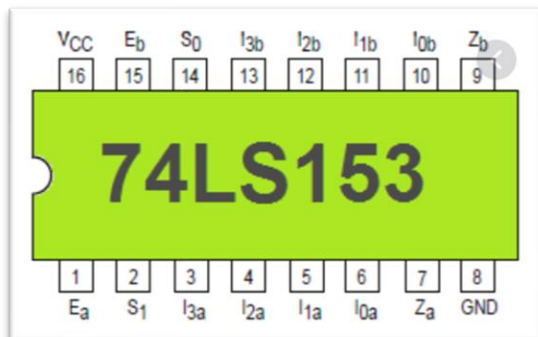
V. DAFTAR PUSTAKA

Mandiri, Mentari Puspita. 2011. Multiplexer dan Demultiplexer, diakses dari <http://mentaripermadi.blogspot.com/2011/12/multiplexer-dan-demultiplexer.html>, pada 26 November 2020

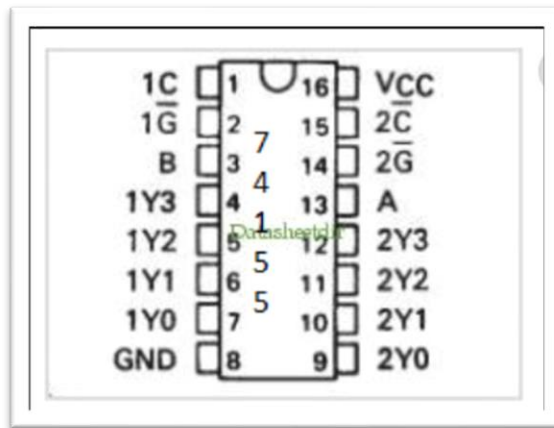
Muhammad Alwiza Ansyar. Saya adalah seorang mahasiswa yang berasal dari Bogor. Saat ini, saya sedang menempuh pendidikan di Universitas Sebelas Maret jurusan Informatika.....

VI. LAMPIRAN

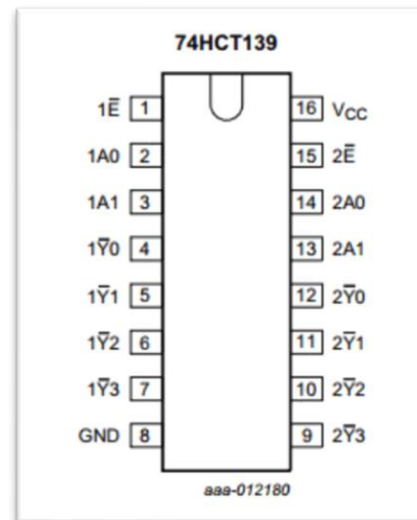
Pinout dari IC 74HC153 (Dual 4-input Multiplexer)



Pinout dari IC 74HC155 (Dual 1x4 Demultiplexer)



Pinout IC 74HC139



Alternatif: menggunakan IC 74139 untuk Demultiplexer 1x4

