

情報科学実験A レポート4

氏名 山久保孝亮
所属 大阪大学基礎工学部情報科学科ソフトウェア科学コース
学籍番号 09B22084
提出日 2024 年 4 月 27 日
担当教員 繁田 浩功/榎井 晃基

1 実験ペアの名前および学籍番号

学籍番号	名前
09B22083	安川雄輝

表 1: ペアの学籍番号と名前

2 送受信回路のステート・マシンの概要図

送受信回路のステート・マシンの概要図は以下の図 1, 図 2 のようになった。

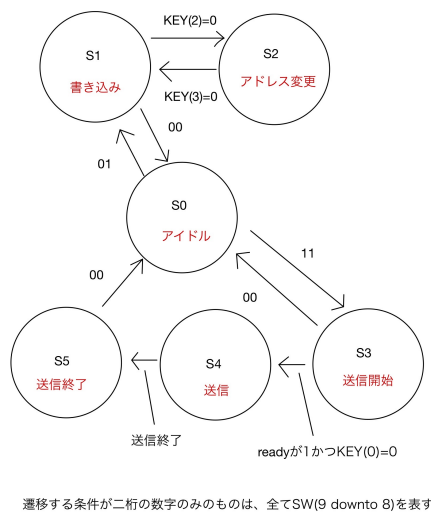


図 1: 送信回路のステート図

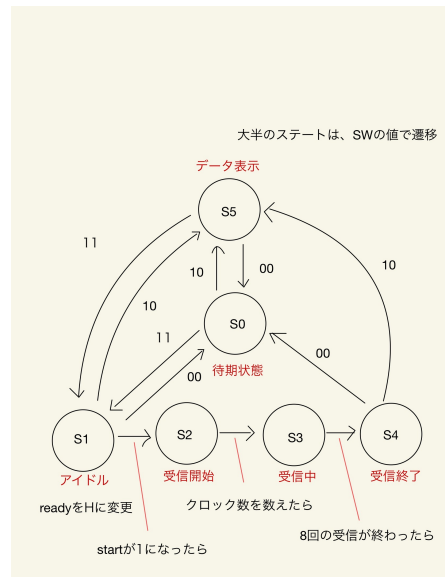


図 2: 受信回路のステート図

3 送受信回路の設計内容

3.1 送信回路の設計

以下では図 1 のそれぞれのステートにおける処理を記述する。

3.1.1 アイドル状態

s0 であるアイドル状態について記述する。32 ビットのベクトルである SW の 8 から 9 ビット目を参照しその値によって次の遷移先を決定する。01 なら s1 に遷移し、11 なら s3 に遷移する。それ以外であれば s0 にとどまり続ける。また、s1 に遷移する際には wadr を 000 に、key2_frag を 0 に、write_in を 0000 に初期化する。これらの変数の具体的な働きは後述する。また、このときに KEY(2) が 0 であるかどうかを判定し 0 であれば key2_frag を 1 としているがこれはテストベンチの KEY の切り替わりをステートの遷移後に計測しているとデータを書き込むのに間に合わなかったので追加した。

3.1.2 書き込み状態

s1である書き込み状態について記述する.KEY(2)が0になっている,即ちKEY(2)が押されるとkey2_fragという変数が1となる.このkey2_fragという変数はKEY(2)が押されたことを記憶しておく変数である.そしてこのkey2_fragが1であればkey2_frag,KEY(3)が押されたかどうかを判定するkey3_fragを0にしてwrite_inに現在のdinの値を格納する.dinはram1の入力であり一度write_inに退避させておくことで次のアドレス変更状態にいる際にデータが変更されても書き込むデータ自体はKEY(2)を押したタイミングでのデータであり続けるからである.そしてその後s2に遷移する.また,s0の時と同様にSWの8から9ビットが00であればs0に遷移し,以上どの分岐にも当てはまらない場合にはs1にとどまり続ける.

3.1.3 アドレス変更状態

s2であるアドレス変更状態について記述する.KEY(2)やKEY(3)が押されるとそれぞれのfragの変数が1となる.key3_fragが1であるかどうかを判定し,1であればkey3_fragを0にしwadrの値が111でなければwadrを1だけ増やす.これにより加算される前のwadrの値に対応するデータの値が確定する.また,wadrの値が111の時はwadrの値を加算しない.詳細は注意点で述べる.その後s1に遷移する.そして最初の分岐に当てはまらなければs2にとどまり続ける.

3.1.4 送信開始状態

s3である送信開始について記述する.SWの8から9ビットが00であればs0に遷移する.次に受信側が送ってくるreadyが1であるかどうかを判定する.readyが1でなければs3にとどまり続け以降の判定は行わない.readyが1のときはKEY(0)が0かどうかを判定する.0であればradrを000に,countを1に初期化してs4に遷移する.

3.1.5 送信状態

s4である送信状態について記述する.ここではまずクロック生成をしてそのクロックの周期を受信側に伝える処理を行う.受信側に伝え終わったときに1となる変数finishが0であるかどうかを判定し,0であればenableを1にしてクロックの生成を開始する.その後生成したclock_txが立ち上りでなければstartを1にして受信側にクロック生成が開始されたことを伝える.また,このとき送信時に使用するクロックの周期をそろえるためにcount_clockというカウンタを用意し,送信側のクロック周期何回分が生成クロックの周期に対応するのか計測しておく.これは送信側と受信側の情報のやり取りをするタイミングをそろえるためである.clock_txが立ち上がりであればstart,enableを0にして周期を伝えることとクロック生成終了する.また,finshを1にして送信状態の次の処理を実行する.ところで,生成クロックの立ち上がりを測定することはclock_tx.lastという直前の生成クロックの値を格納する変数を用意しておき,現在の生成クロックの値が1で,直前の値が0のときという条件分岐を作って実現した.

そのあとすぐにデータの送信を開始する.送信したデータの個数を表すcount_sendという変数が8かどうかを判定することで7以下である場合は同じ処理が繰り返されるようにした.7以下である場合は送信側のクロックが何回立ち上がったかを記憶するclockという変数が先ほど計測したcount_clockから1引いた値と同じになったタイミングでデータを送信する.今回送信するデータと実機の7セグメントデコーダの対応は以下の通りである.

7セグメントデコーダの番号	送信するデータ
HEX0	常に0を表示
HEX1	送信するデータ
HEX2	常に0を表示
HEX3	送信するデータのアドレス
HEX4	常に0を表示
HEX5	ready 信号

表 2: 7セグメントデコーダの送信するデータの対応表

3.1.6 送信終了状態

s5である送信終了状態について記述する.SWの8から9ビットが00であればradr,clk_tx_last,count,count_send,count_clkを初期化してs0に遷移する.それ以外の場合はs5にとどまり続ける.

3.2 受信回路の設計

以下では図2のそれぞれのステートにおける処理を記述する.

3.2.1 待機状態

s0である待機状態ではSWの8から9ビットの値によってstateを遷移させる.11のときはs1に遷移し10のときはs5に遷移する.

3.2.2 アイドル状態

s1であるアイドル状態ではready信号を1にして送信側のstartが1になるのを待つ状態である.送信側のstart信号が1になるとs2に遷移し,SWの8から9ビットの値が00であればs0に,10であればs5に遷移する.s2に遷移するときはwadrを000に,生成クロックの周期を計測するための変数countを1に初期化する.これはclkが次に立ち上がったときに数え始めることになってしまうためである.

3.2.3 受信開始状態

s2である受信開始状態では生成クロックの周期の計測をする状態である.startが0になるとs3に遷移する.

3.2.4 受信中状態

s3である受信中状態ではcountの回数になるまでクロックの立ち上がり数を数える変数iを増やしていく.countとiの値が等しくなった時にwadrをインクリメントする.書き込みアドレスが最大値111まで達するとs4に遷移する.

3.2.5 受信終了状態

s4である受信終了状態ではSWの8から9ビットの値によってstateを遷移させる.SW8から9ビットの値が00の時はs0に,10のときはs5に遷移する.

3.2.6 データ表示状態

s5 であるデータ表示状態では count_read によって 500000000 回クロックの立ち上がりを計測するたびに radr の値を 1 ずつ増やしていった受信したデータを表示させる。最後までデータを衆力し終わると radr の値を初期化して s5 にとどまり続ける。SW8 から 9 ビットの値が 00 の時は s0 に、11 のときは s1 に遷移する。

4 テストベンチによる動作確認内容・結果

テストベンチは配布されていたものをそのまま利用した。

4.1 送信回路の動作結果

4.1.1 データ書き込みの動作

以下の図 3 はデータ書き込みを行う際の波形である。以下でその詳細を記述する。

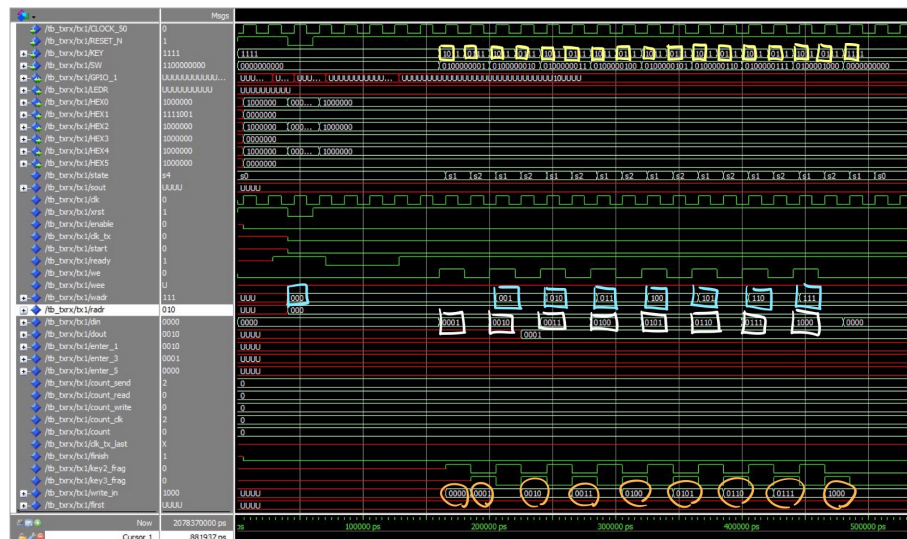


図 3: データ書き込みの際の波形

黄色く囲まれた一番上の段の一番左のビットである KEY(3) が 0 になったときに状態が s2 に遷移し、白く囲まれた直前の状態である s1 のときの din の値がオレンジで囲まれた write_in にコピーされていることがわかる。また、黄色で囲った左から二番目のビットにより KEY(2) が変更されると状態が s1 へ遷移し、水色で囲んだ wadr の値が 1 増やされていることがわかる。これらの操作は合計で 8 回行われているため 8 回分のデータを書き込むことができています。

4.1.2 データ送信の動作

以下の図 4 はデータ送信時の波形である。以下でその詳細について述べる。

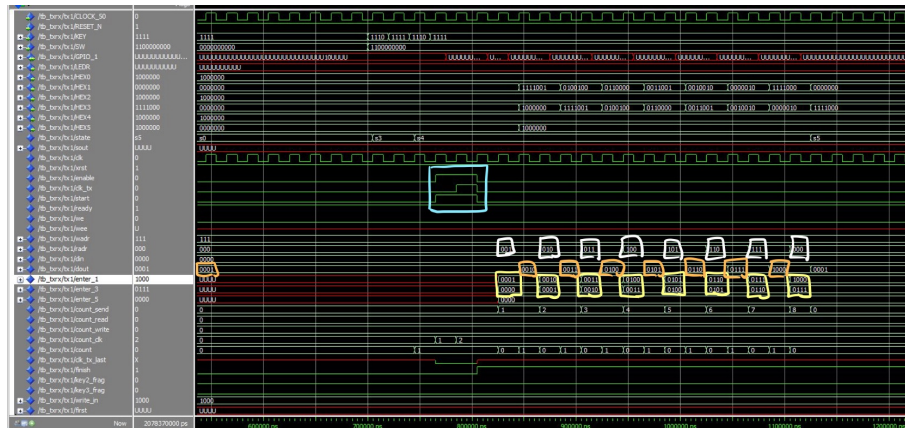


図 4: データ送信の際の波形

水色で囲んだ部分から状態が s4 に遷移した後に enable と start が同じタイミング 1 となっている。また、この期間は clock.tx の一周期分となっているため正しい。また白色に囲んだ部分から一つ前の radr の値が黄色で囲んだ enter_3 に、オレンジに囲んだ部分から一つ前の dout の値が黄色で囲んだ enter_1 に入っている。これは先ほど書き込んだデータを同じアドレスで出力できている。

4.2 受信回路の動作結果

4.2.1 データ受信の際の波形

以下の図 5 はデータ受信時の波形である。以下でその詳細について述べる。

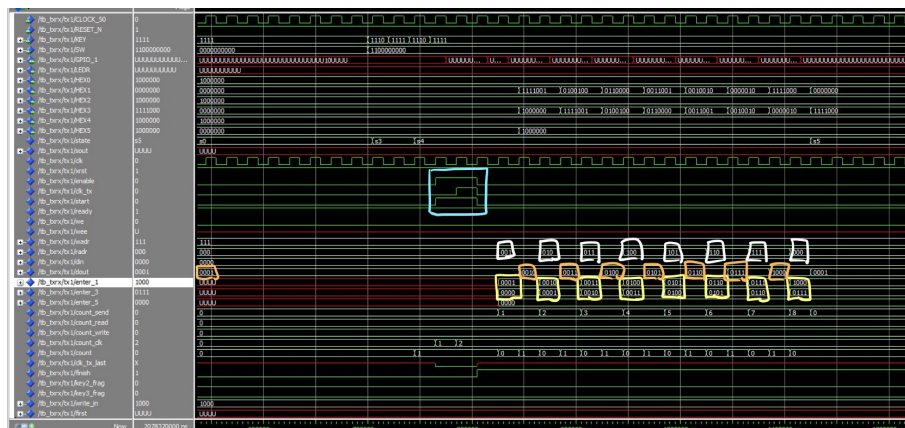


図 5: データ受信の際の波形

水色で囲んだ部分では start が 1 に立ち上がってから count でその周期を計測できていることがわかる。白で囲んだ部分からアドレスが、オレンジで囲んだ部分から受信したデータがそれぞれ黄色で囲んだ enter_2 と enter_3 に順に格納されている。この操作は 8 回行われているので正しく動作している。

5 非同期通信回路作成における注意点などの考察

今回の課題を通して非同期通信回路を作成するにあたって注意しなければならない点は以下の 2 点であると考えた。

1. クロックを生成し終えてからデータを送信するまでのタイミングを合わせること.
2. アドレスが 000 に戻ったときに 000 に格納されているデータが変化してしまう.

以下でその詳細について述べる.

1. 今回の実験の仕様から,KEY(2) が押されると SW の下位 4 ビットの値の送信を開始する状態に遷移し, そのあと KEY(3) を押すとクロックを生成する. そして最後にデータを送信するが, クロック生成を終えてからデータを送信する間を制御するボタンが設定されていない. 今回の課題では設計内容で記述した通りクロック生成が終了した後すぐにデータを送信したが, この部分は送信側と受信側できちんとあらかじめ決めておく必要がある. 今回の実験では周期をお互い決めると即座に送受信を開始するという仕様に決めた.
2. RAM にデータを書き込む際は書き込むためのアドレスの値が変化するとき, 最後に入力に入っていた値が格納される. 即ちアドレスが変更された瞬間に変更される前のアドレスに格納されるデータが確定し, 現在のアドレスが指す RAM の内容は常に変動しているということである. 今回のアドレスは 3 ビットのベクトルを 1 ずつ増やしていくことでアドレスを実現したが,111 の状態で 1 を加算すると 000 に戻ってしまっていた. したがって, 最初に 000 のアドレスにデータを格納した後 111 までデータを格納すると, アドレスが 000 まで戻ってしまうため 000 のアドレスに格納されている値が変化してしまう. したがって,wadr が 111 のときはアドレスが加算されないようにすることで前述の問題が発生しないように工夫した.

6 教員・TA による動作確認時刻

1 月 25 日 18 時 15 分